122400

R-685

CHARLES STARK DRAPER LABORATORY

A Division of Massachusetts Institute of Technology • Cambridge, Mass

(NASA-CR-122400) PRELIMINARY SOFTWARE DEVELOPMENT FOR OPTICAL MIRROR FIGURE CONTROL Technical Report, May - Dec. 1970
D. MacKinnon (Massachusetts Inst. of Tech.)
Dec. 1970 114 p
CSCL 14B G3/14
N72-24488
Unclas
28137

PRELIMINARY SOFTWARE DEVELOPMENT FOR OPTICAL MIRROR FIGURE CONTROL

MAY 1972
RECEIVED
NASA STI FACILITY
INPUT BRANCH

DUNCAN MAC KINNON

R-685

PRELIMINARY SOFTWARE DEVELOPMENT FOR OPTICAL MIRROR FIGURE CONTROL

by

Duncan MacKinnon

December 1970

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CHARLES STARK DRAPER LABORATORY
CAMBRIDGE, MASSACHUSETTS 02139

This report covers research conducted between May 1970 and December 1970

Approved: _____
Assistant Director

Approved: _____
Deputy Director

ACKNOWLEDGMENT

The author wishes to acknowledge the sponsorship of Hector Ingrao of the National Aeronautics and Space Administration whose interest in the application of modern control and structural theory to active primary mirrors led to the establishment of the project reported herein. The author also expresses his thanks to John Mangus of Goddard Space Flight Center and Charles Jones of the Marshall Space Flight Center for reviewing this report and contributing a number of valuable suggestions.

The author wishes to thank Eugenia Freiburghouse for her assistance in preparing the programs and program descriptions presented in this report.

This report was prepared under DSR Project 55-34900 sponsored by the Goddard Space Flight Center of the National Aeronautics and Space Administration through contract NAS 5-21542.

The publication of this report does not constitute approval by the National Aeronautics and Space Administration of the findings or the conclusions contained therein. It is published only for the exchange and stimulation of ideas.

R-685

PRELIMINARY SOFTWARE DEVELOPMENT FOR OPTICAL MIRROR FIGURE CONTROL

ABSTRACT

The maintenance of accurate primary mirror figure in the face of environmental disturbances is the key to the achievement of diffraction-limited performance in a large space telescope. In order to develop the concepts of optical mirror figure control, an experimental program has been initiated at the Marshall Space Flight Center, Huntsville, Alabama. A major component in this experiment will be an XDS Sigma 5/7 digital computer which will realize the control algorithm. This report presents a description of a software package which realizes linear optimal, simplified linear and iterative optimal control algorithms. The software, in addition, provides for interactive communication between the operator and the computer, and interaction between the computer and the experimental hardware elements. A brief description of a small hybrid computer system is also presented.

by Duncan MacKinnon
December 1970

TABLE OF CONTENTS

TABLE OF CONTENTS (cont)

EXPERIMENTAL ACTIVE MIRROR SOFTWARE COMPONENTS

SOFTWARE COMPONENTS
VARIABLE LIST

EXPERIMENTAL ACTIVE MIRROR PACKAGE

AIM	INVERSE OF AM, TEMPORARY STORAGE
AM	DEFORMATION-FORCE MATRIX, TEMPORARY STORAGE
ARM	REDUCED VERSION OF AM
ARRM	REDUCED VERSION OF ARM
ASCALV	ACTUATOR COMMAND SCALE FACTORS
DA	TEMPORARY STORAGE FLOATING POINT
DTITCS	ACTUATOR SET TIME FOR ITERATIVE CONTROL SYSTEM
DUMV	TEMPORARY STORAGE VECTOR
FSCALV	FIGURE SENSOR MEASUREMENT SCALE FACTORS
GAINM	CONTROL SYSTEM GAIN MATRIX
GAINV	CONTROL SYSTEM GAIN VECTOR
I	INDEX
IA	TEMPORARY STORAGE INTEGER
ICHNG	SWITCH FOR MODIFYING VARIABLE VALUE
J	INDEX
K	INDEX
L	INDEX
LACTV	ACTUATOR LOCATION DESIGNATOR
MODE	TYPEWRITER CONTROL MODE
MODES	DESIRED VALUE OF MODE
MODOP	DESIRED CONTROL SYSTEM OPERATING MODE
MODOPS	DESIRED VALUE OF MODOP
N	NUMBER OF FIGURE MEASUREMENTS
NR	NUMBER OF ACTUATORS
NRA	DIMENSION OF GAINM
NSNSWT	NUMBER OF SENSE SWITCH USED TO INTERROGATE TYPEWRITER
TREAL	REAL TIME VARIABLE
UFAV	MEASURED ACTUATOR OUTPUTS
UFV	ACTUATOR INPUTS
XFRV	REDUCED XFV VECTOR
XFSV	X COORDINATES OF MEASUREMENT POINTS
XFV	MEASURED FIGURE ERRORS
XV	MAIN DATA BASE
YFSV	Y COORDINATES OF MEASUREMENT POINTS

PARAMETER OPTIMIZATION PACKAGE

EPS	STEP SIZE
GRADV	GRADIENT VECTOR
NHC	NUMBER OF HALVINGS COMPLETED
NHM	MAXIMUM NUMBER OF STEP SIZE HALVINGS
NIC	NUMBER OF ITERATIONS COMPLETED
NIM	MAXIMUM NUMBER OF SUCCESSFUL ITERATIONS
NITPRT	PRINT OUTPUT EVERY NITPRT ITERATIONS
NPAR	NUMBER OF ADJUSTABLE PARAMETERS
PARV	ADJUSTABLE PARAMETER VECTOR
PINDEX	PERFORMANCE INDEX VALUE
PISTOR	PERFORMANCE INDEX AT END OF LAST SUCCESSFUL ITERATION
SLGV	STORED VALUES OF GRADIENT VECTOR LENGTH
SPIV	STORED VALUES OF PERFORMANCE INDEX

CHAPTER 1
AN EXPERIMENTAL ACTIVE MIRROR

1.0 Introduction

Astronomical observations through a large earth-based telescope suffer from limitations placed on the resolving power of the telescope by fluctuations in the earth's atmosphere. As part of a space orbiting astronomical laboratory, however, a telescope would not be subject to these limitations. A large instrument which is diffraction - limited over a major part of its useful spectrum of observation was envisioned. Maximum resolving power requires extremely accurate maintenance of the figure of the primary mirror.[1,2]

Although it was possible to suitably polish such a large mirror to the desired surface accuracy, stresses introduced by thermal variations in the mirror and fluctuations in support structure loads, structural instability and the elimination of gravity loading in orbit could create surface perturbations which would exceed the surface accuracy limits required for diffraction limited performance. As a result, investigators have attempted to develop techniques for actively correcting the mirror figure in a space environment and a number of promising control techniques have been developed.[1-8] The development and application of these control concepts is one of the key challenges facing the designer of the large space telescope.

While the large scale digital computer is an extremely valuable tool for the analysis and simulation of complicated mirror figure control systems, the results obtained are only as reliable as the modelling accuracy of the physical components in the system. Accurate modelling requires a considerable amount of intuition if the trade-off between modelling

accuracy and computation time is to be satisfactorily resolved. Often, terms neglected in the modelling process are of key importance to the overall system design.

To resolve these problems it is important to have some way of checking the results of numerical analysis against actual system behavior. Such checks are furnished by an experimental program.

Experimental work in the past has been largely conducted using analog devices to synthesize figure actuator commands from surface error measurements. As a result of the expense and time associated with programming a general purpose analog computer or constructing a special purpose analog system, it has been difficult to explore the full spectrum of control solutions or efficiently process experimental data.

In response to these problems, a more efficient experimental tool has evolved in the hybrid digital analog computer system. Spurred by declining cost hybrid computing systems are appearing in a wide variety of laboratory environments. This report describes the application of a hybrid computing system to an experimental active mirror and the required software packages.

1.1 Experiment Design Concept

The experimental system consists of the primary mirror fitted with actuators for figure modification, a mirror figure sensor and an "on line" control system processing the figure errors measured by the sensor to provide proper corrective signals to the actuators.

Following a modern approach a digital computer has been selected as the control processor of the experimental system. The utilization of a general purpose computer has a number of advantages, including:

1. Programmability permitting a large number of different primary mirror control configurations to be investigated without extensive hardware modification.
2. The ability to handle a number of auxilliary tasks such as experimental data processing and display.
3. Characteristics similar to the system computer which will be used in an orbiting astronomical observatory.

1.2 A Computer Controlled Active Mirror

A block diagram of an experimental active mirror is shown in Figure 1.2.1. The digital computer consists of a central processor, a random access core memory and an input-output processor. The central processor handles arithmetic operations, logic operations and some date transfer operations using instructions extracted from the core memory. The input-output processor controls the transfer of information from the central processor and core memory to computer peripherals which are part of the interface with the real world. The core memory holds two types of stored information, program instructions and program storage. The program instructions tell the computer what to do with information extracted from program storage and input devices.

The interface consists of a number of components. The digital output buffers are a set of addressable registers which temporarily store digital information which is used by devices such as the external switch

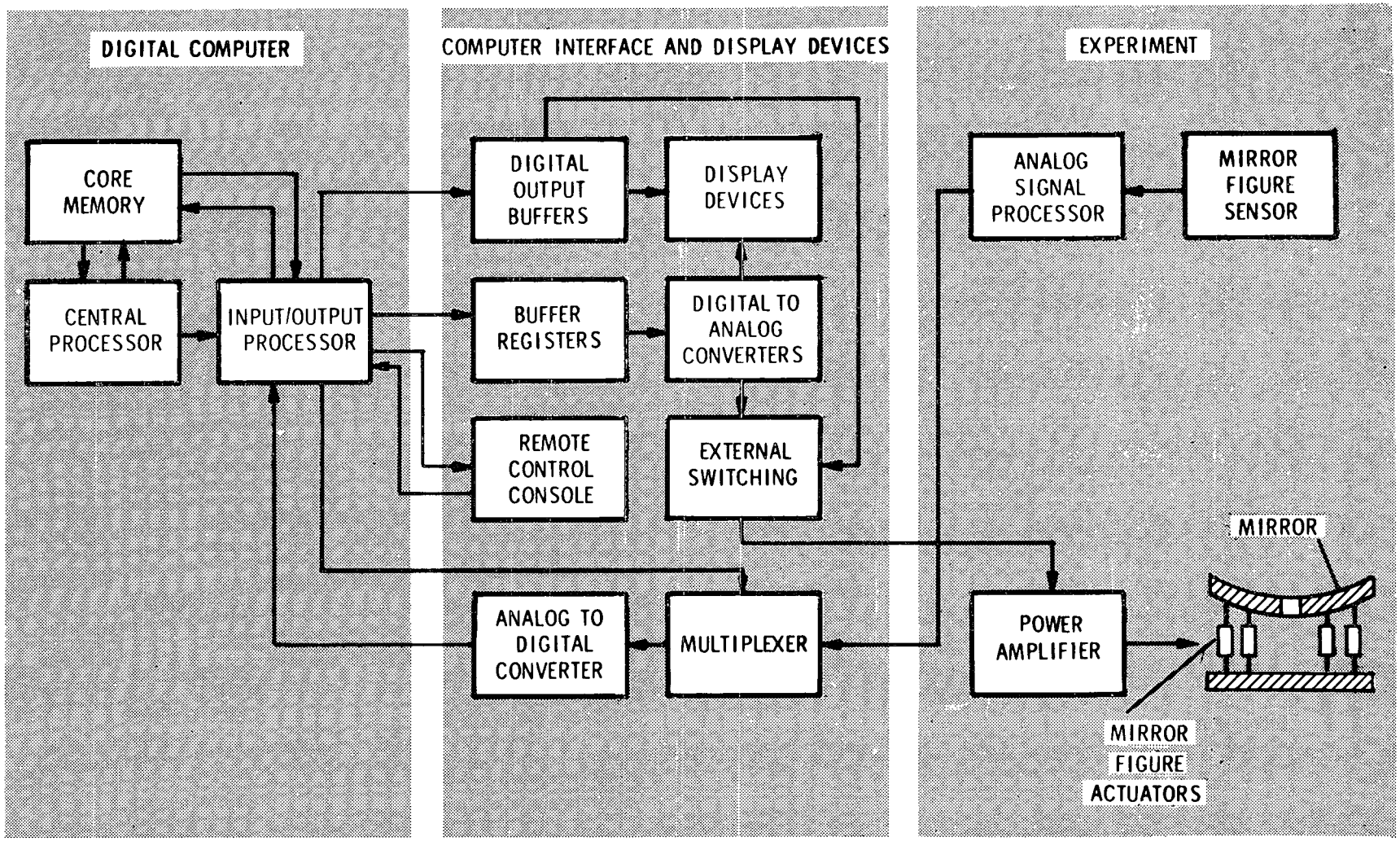

Figure 1.2.1 A digital control system for an experimental active mirror.

and data displays. The buffer registers temporarily hold digital data which is converted to an analog signal by the digital to analog converters. The output signals from the digital to analog converters are inputs to display devices such as meters and scopes and an external switch which expands the capacity of the converter by permitting a larger number of channels to be addressed at a lower switching rate. The external switch is controlled by digital signals from the digital output buffer. Analog signals to be entered into the computer pass through a computer-controlled switch or multiplexer; the single output of the multiplexer is converted to a digital signal by the analog to digital converter. The digital signal is then transferred to the central processor or the correct location in the core memory by the input-output processor.

External to the computer and its interface is the mirror figure sensor. Signals from the figure sensor pass through an analog signal processor, if required, to the multiplexer. The outputs from the external switch pass to a power amplifier which supplies the energy required to drive the mirror figure actuators.

PRECEDING PAGE BLANK NOT FILMED

CHAPTER 2

GENERAL FEATURES OF THE EAM SOFTWARE PACKAGE

2.0 General Specifications

The software package is divided into 4 major sections:

1. EAM Package for Linear Optimal or Simplified Linear Figure Control (EAM).
2. Parameter Optimization Package for Iterative Figure Control (POP).
3. Mathematical Operations Package (MOP).
4. Input Output Operations Package (IOP).

The EAM package contains programs which realize the Linear Optimal (LOCS) and Simplified Linear (SLCS) control laws. The programs also calculate system gains and initialize system parameters. An operating sequence monitoring capability is also provided which prevents operator instruction sequence errors. The programs also permit the operator to operate and diagnose the EAM remotely using a typewriter console. The subroutines also realize servo loops for the actuators. The addition of actuator output control loops reduces the sensitivity of the control system to variations in actuator scale factors due to nonlinearity and structural instability.

The Parameter Optimization Subroutine Package provides the software for the Iterative Figure Control System (IFCS). The gradient of the performance index

$$J = x_f' x_f \qquad (2.0.1)$$

where x_f is the measured figure error vector

is generated by perturbing the elements of the actuator servo input vector. The resultant gradient information is used in a steepest descent procedure to generate a sequence of actuator outputs which approach the linear optimal control solution. Modified versions of the steepest descent algorighm featuring improved rates of convergence may also be selected by the operator.

The Mathematical Subroutine Package is used by all the EAM programs to perform mathematical operations such as matrix addition, subtraction, multiplication and inversion, etc. All matrices are stored as single dimensioned arrays. While this complicates the programming somewhat the problem of carrying dimensions through subroutines and restrictions on variable dimensioning are eliminated.

The Input-Output Subroutine Package has been developed at the Charles Stark Draper Laboratory to permit highly efficient handling of input data and output print in terms of core storage for program instructions. Further savings are realized by reading and printing headings associated with input data. This procedure eliminates many format statements which would otherwise require a large amount of core.

2.1 Computer-Operator Communication

The computer exists in an isolated electronic world. All communication with the external environment must take place through channels which process the information in a form suitable for computer or peripheral use. The data channels include analog to digital converters and digital information buffers for input and digital to analog converters and analog buffers for output. Two levels of software are envisioned for effective communication.

The first division is a machine oriented set or programs which perform data transfer on the lowest channel level. Such software is written in machine language and provides an interface between the higher level FORTRAN programs and computer hardware. For example, if it was desired to plot information on an analog X-Y plotter the computer programmer would call a FORTRAN program PLOT which would scale the data and provide interpolated values if necessary. The resultant scaled data would be transfered to the analog channels assigned to the X and Y coordinates by a machine language subroutine which would receive the channel numbers and the data from the FORTRAN program. Such a procedure offers a maximum amount of flexibility since PLOT in FORTRAN is easily modified or expanded while the machine oriented subroutine can remain fixed. In addition, conversion to a new computer facility can generally be accomplished with minor changes in the FORTRAN routines and the substitution of a new set of machine language subroutines.

The EAM software is designed to take maximum advantage of the characteristics of a variety of available computer peripherals. Typical input-output peripherals may be classified as shown in table 2.1-1. The line printer, high speed card reader and magnetic tape are ideal devices for transferring large blocks of data. A typewriter or teletype, on the other hand, is best suited for making modifications to or extracting small blocks of data from the program data base.

2.2 The Data Buss Concept

In order to assure maximum access to program storage a "data buss" has been established. The "data buss" is a one dimensional array designated XV. All program variables are equivalenced to sections

of XV. XV is transmitted throughout the EAM software by means of a labelled common block BLKMFC. Transmission of information via COMMON storage eliminates the requirement for long parameter lists in subroutine calls.

Table 2.1-1 Peripheral device characteristics.

DEVICE	DATA TRANSFER RATE		COST		SIZE	
	LOW	HIGH	LOW	HIGH	SMALL	LARGE
LINE PRINTER		✓		✓		✓
HIGH SPEED MAGNETIC TAPE		✓		✓		✓
HIGH SPEED CARD READER		✓		✓		✓
TYPEWRITER OR TELETYPE	✓		✓		✓	

CHAPTER 3

DESCRIPTION OF PRINCIPAL EAM SOFTWARE COMPONENTS

3.0 Introduction

The following sections of Chapter 3 are general descriptions of the major components of the EAM software package. User descriptions are presented in Chapter 4 while program listings are contained in Appendices A, B, C, and D. The software is written using a combination of XDS Fortran II and IV and is designed for execution on an XDS Sigma 5/7 computer.[9-11]

Program and subprogram names (MAIN, TYPCON, ACTCAL, etc.) and program variables (MODOP, MODE, XV, etc.) are capitalized in the text. Indexed variable names are usually followed by the index in brackets (XV(I), UFV(I), etc.).

3.1 Main Program for the Experimental Active Mirror

Program MAIN provides the skeletal structure for the EAM. The program initially interrogates the typewriter through TYPCON. Control remains with the typewriter until the control system operating mode (LOCS, SLCS, or IFCS) is defined. At that point the defined operating mode (MODOP) is stored and MAIN monitors the sequence of typewriter commands to the system. The correct operating sequence is:

1. Identify control configuration (MODE = MODOP > 5).
2. Initialize EAM (MODE = 1).
3. Start EAM (MODE = 2).

Operation of the EAM control system may be terminated at any time by setting sense switch NSNSWT (defined by input data) which transfers

control back to the typewriter. The typewriter may then be used to examine or change any piece of information in the DATA BUSS. When control is returned to MAIN the status of MODE and MODOP is checked to ensure that they have not been altered. Alteration generates a typed operator error message and returns control to the typewriter for correction. A flow diagram of the main program is shown in Fig. 3.1.1.

3.2 Typewriter Control of the EAM

Since the computer is generally some distance from the experiment it is desirable to provide a remote control capability adjacent to the experiment. In the current configuration this capability is provided by a typewriter which communicates with the EAM software through the subroutine TYPCON.

When TYPCON is called with NENTRY = 2 the typewriter is interrogated for a value of MODE. MODE can have integer values between 1 and 10. The operations corresponding to these modes are

MODE = 1 — Initialize the Mirror Figure Control System (MFCS) Control is transferred to MAIN which reads in data and initializes the EAM providing MODOP has been defined. If MODOP was not defined an error message is typed and new value of MODE requested by TYPCON.

MODE = 2 — Start the Mirror Figure Control System. Control is transferred to MAIN which starts the control system providing the previous value of MODE was 1 and MODOP has not changed since MODE was equal to 1. If these conditions are not satisfied an error message is typed and a new MODE requested.

MODE = 3 Diagnostic mode. The abbreviated name of a variable in the DATA BUSS is requested. The typed reply is tested for conformity with the cataloged names. If the reply is not in the catalog an error message is typed and a new name requested. If the name is accepted, it is typed. The subscript(s) of subscripted variables are then requested. The operator types in the subscript(s) which is retyped by the computer. TYPCON then interrogates the value of the identified parameter and types it. A new variable name is then requested. Diagnosis may be continued at this point by typing in a variable name or terminated by typing DEND. If DEND is typed TYPCON will request a new value for MODE. To save time and storage space the variable names are abbreviated as indicated in Table 3.2.1.

MODE = 4 Test actuators. Subroutine ACTCAL is called and the actuator scale factors printed on the line printer. The end of ACTCAL is signaled by a typed message and a new value for MODE is requested.

MODE = 5 Test mirror. Subroutine MIRCAL is called and the reduced deformation - actuator input matrix generated. The resultant matrix is printed and the end of calibration indicated by a typed message and the request for a new MODE.

MODE =6 Simplified linear figure control. TYPCON sets MODOP = 6 and transfers control to MAIN. This operation defines a simplified linear control mode which may then be initialized and operated.

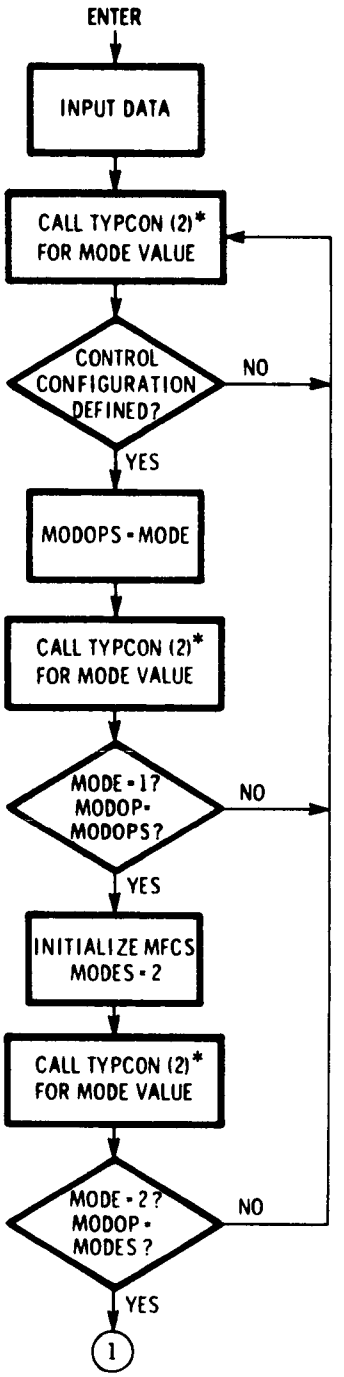

Fig. 3.1.1 Main program for the experimental active mirror (MAIN).

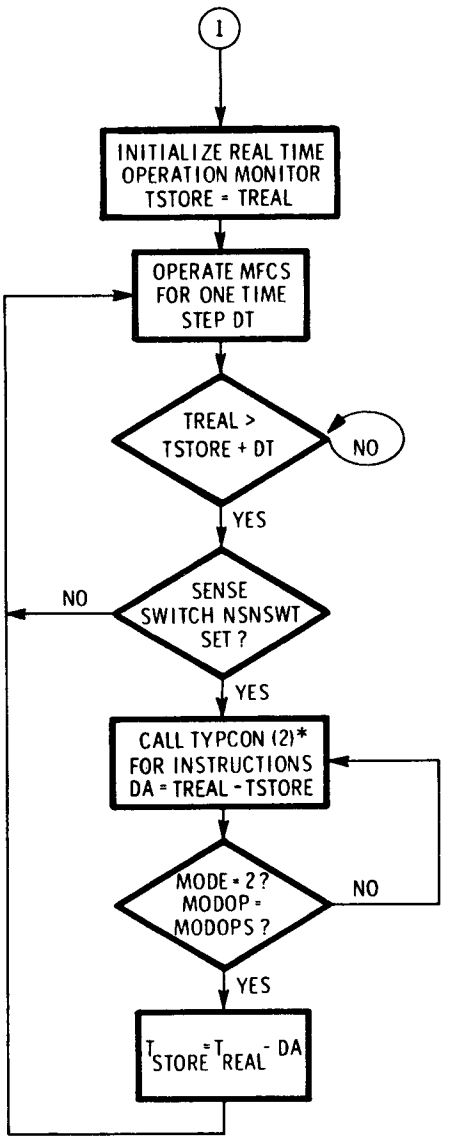

Fig. 3.1.1 Main program for the experimental active mirror (MAIN). (Cont)

Table 3.2.1 Program variable abbreviations

PROGRAM VARIABLE	ABBREVIATED NAME
AIM	AIM
AM	AM
ASCALV	ASCV
DUMV	DUMV
FSCALV	FSCV
GAINV	GANV
GAINM	GANM
LACTV	LACV
UFAV	UFAV
UFV	UFV
XFRV	XFRV
XFSV	XFSV
XFV	XFV
XV	XV
YFSV	YFSV

MODE = 7 Linear optimal figure control. TYPCON sets MODOP = 7 and transfers control to MAIN. MODOP = 7 specifies a linear optimal control mode which may then be initialized and operated.

MODE = 8 Iterative figure control. MODOP is set equal to 8 by TYPCON. MODOP = 8 defines an iterative control mode which may then be initialized and operated.

MODE = 9 Modify data buss value. This mode establishes a condition in which it is possible for the operator to modify any element in the data buss. The element is identified and indexed using the procedure under MODE = 3. A new value for the completely identified variable is then accepted; the variable value changed and the resulting value printed.

A flow diagram of TYPCON is shown in Fig. 3.2.1.

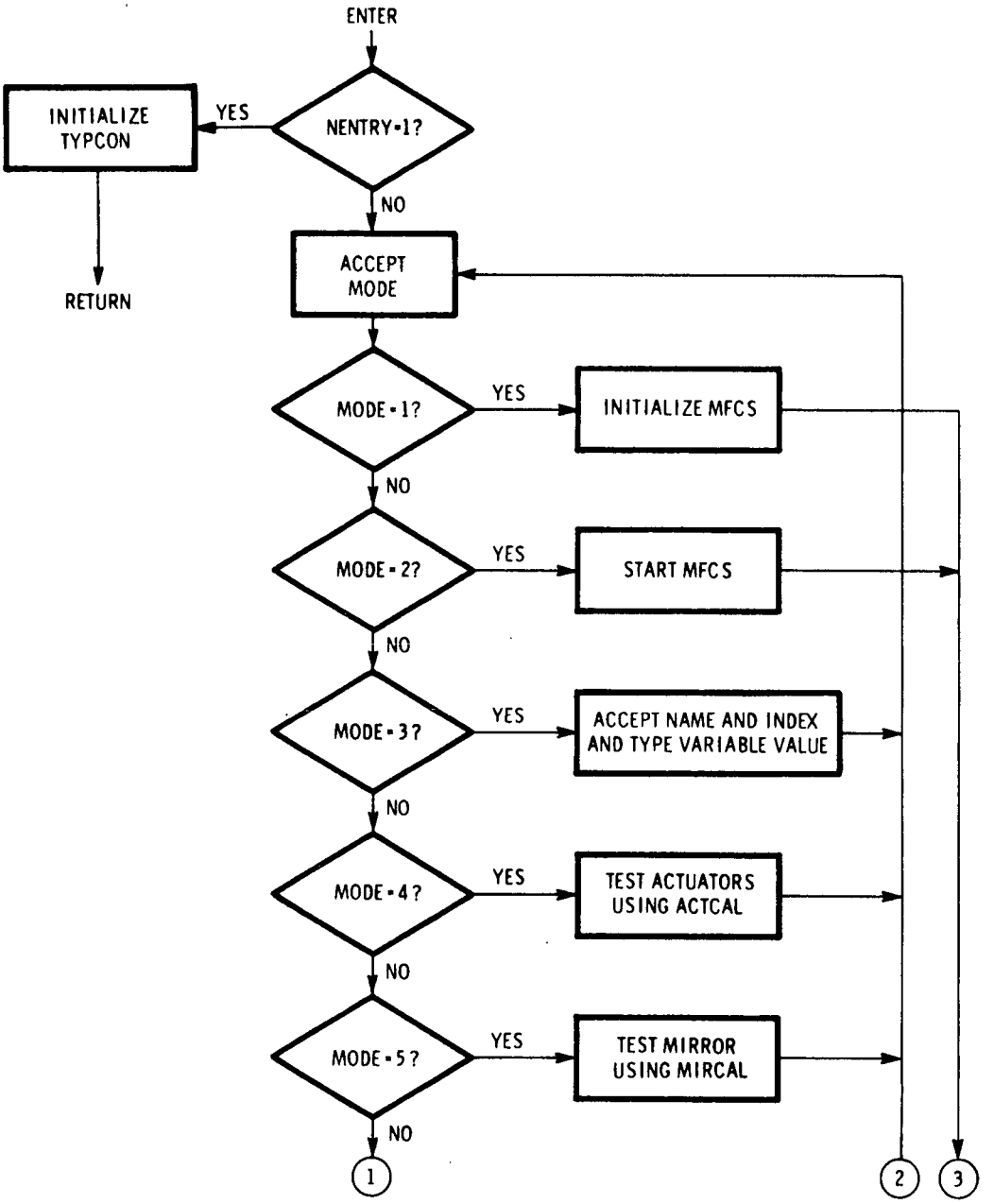

Fig. 3.2.1 Flow diagram of the typewriter control algorithm for the experimental active mirror (TYPCON).

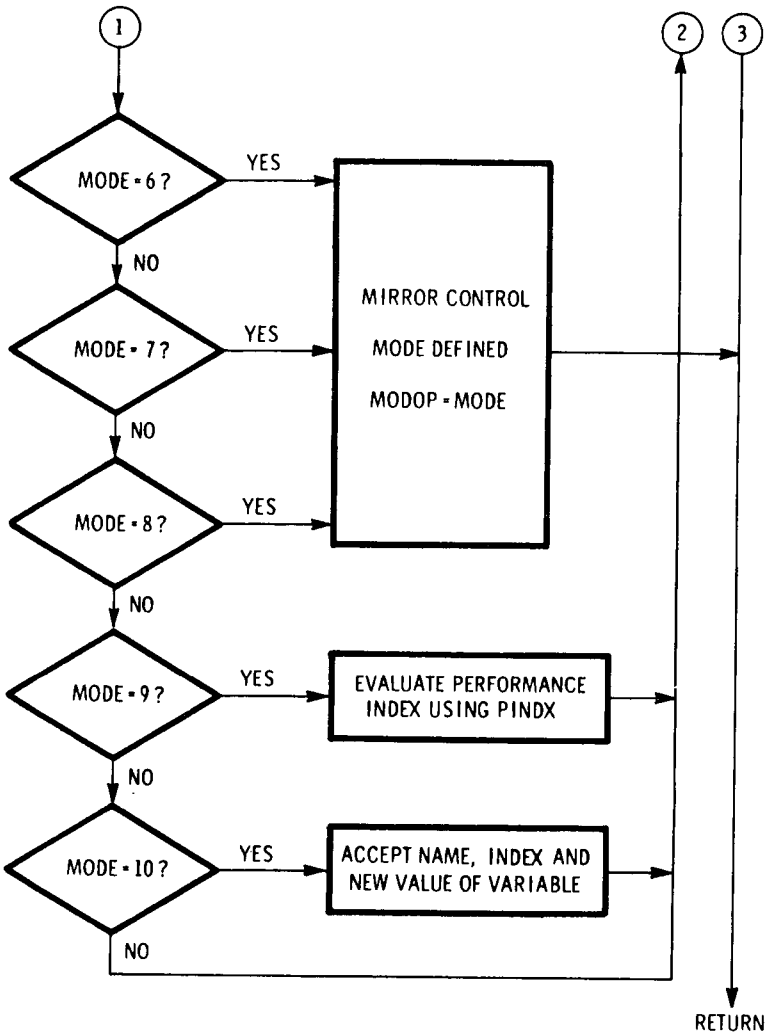

Fig. 3.2.1 Flow diagram of the typewriter control algorithm for the experimental active mirror (TYPCON). (Cont)

3.3 Experimental Active Mirror Control Algorithm

The mirror figure control algorithm is realized in subroutine MFCS. When the parameter NENTRY = 1 the gains associated with the control law and the mirror actuators are read in and the feedback gain matrices for the Linear Optimal Control System or Simplified Linear Control System calculated. If Iterative Figure Control is desired the main parameter optimization program MINFCN is initialized. If NENTRY = 2 the program calculates the linear optimal or simplified linear actuator commands. If NENTRY = 4 MFCS transfers the commands calculated by the Iterative Figure Control System to the actuator servos. A flow diagram of MFCS is shown in Fig. 3.3.1. The computational flow of the Iterative Figure Control System (MINFCN) is shown in Fig. 3.3.2. A detailed discription of the algorithm is presented in Ref. 1.

3.4 Experimental Active Mirror Initialization

The basic initialization of the EAM is accomplished by subroutine INIT. The initialization operations depend on the value of the parameter NENTRY which is determined when INIT is called. If NENTRY is equal to 1 INIT reads in the basic parameters associated with the EAM hardware components. When NENTRY is equal to 2 INIT initializes the program variables such as figure actuator output, input, and the figure measurement vector associated with the figure sensor.

3.5 Actuator Command Processing

Subroutine ACTCMD is called with an integer parameter I which identifies the Ith actuator. The subprogram interrogates the Ith actuator output sensor through machine language coding and transfers the reading to the calling program using UFAV(I). The Ith actuator input command UFV(I) is then multiplied by a scale factor ASCALV(I) which permits

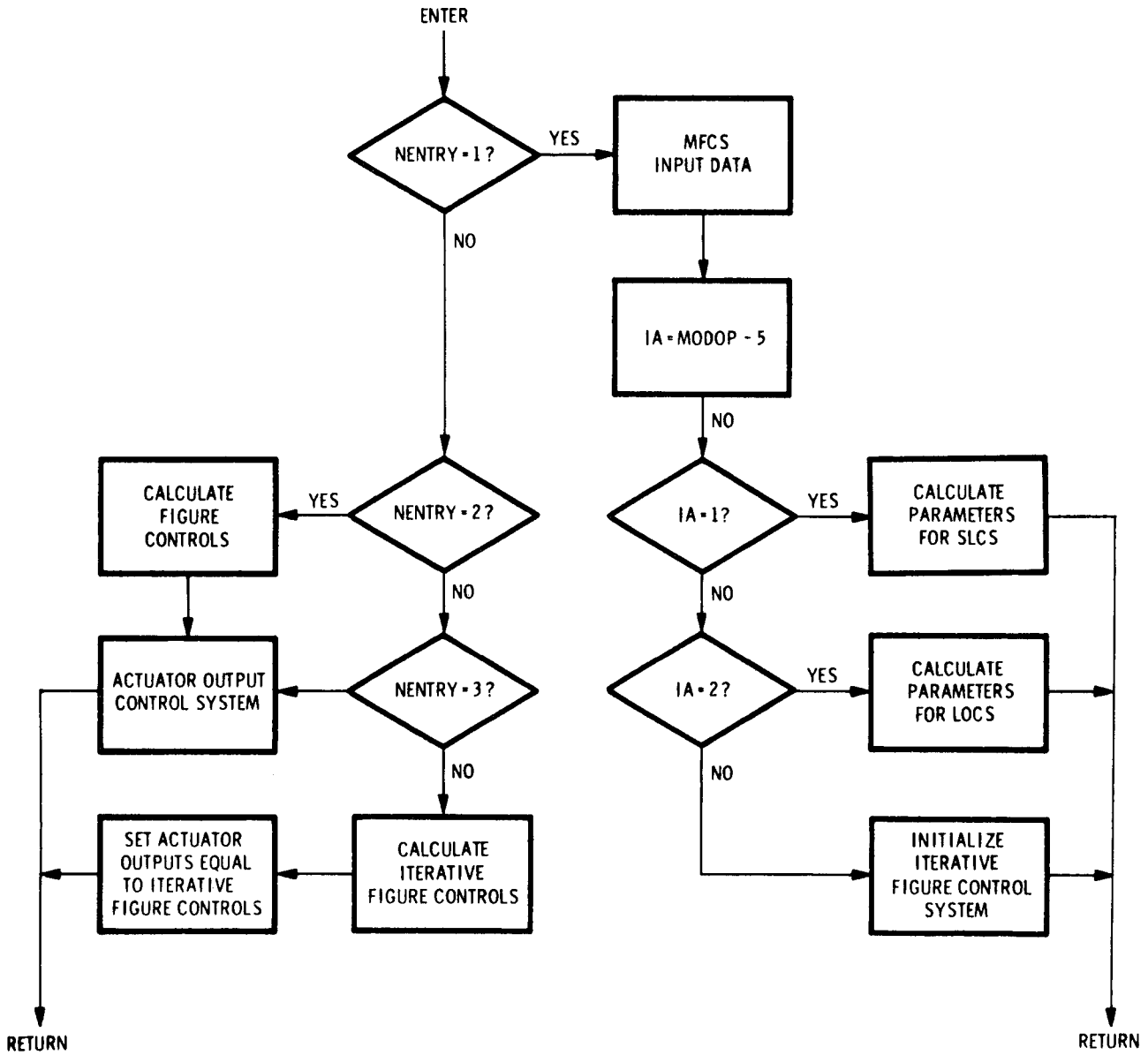

Fig. 3.3.1 Flow diagram of mirror figure control system subroutine (MFCS).

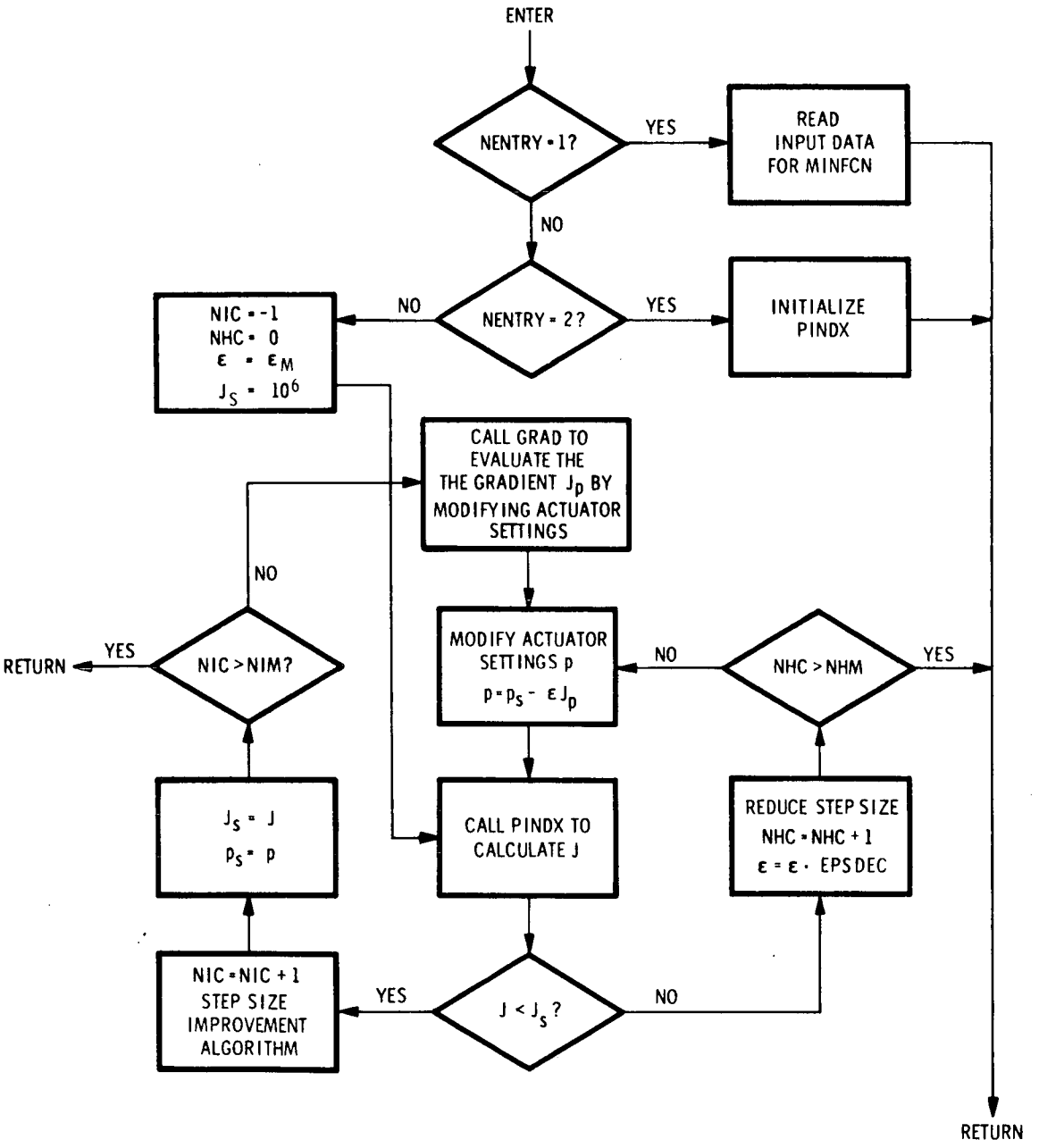

Fig. 3.3.2 Flow diagram of the iterative figure control system (MINFCN).

compensation of scale differences between different actuators. The scaled actuator command is transferred to the actuator power modulator by a set of machine language statements which communicate with the D/A or D/D actuator channels. The subroutine features built-in adjustable time delays which allow sufficient time for the interface components to operate.

3.6 Figure Sensor Measurement Processing

Figure error measurements are obtained by the computer using subroutine FIGSEN. FIGSEN supplies the X and Y coordinates (XFSV(I), YFSV(I)) of the I' th figure error XFV(I) to the figure sensor image dissector. The figure sensor error measurement is returned to FIGSEN and multiplied by the scale factor FSCALV(I). In order to reduce the effects of sensor noise the measurements are repeated NMEAS times and the results averaged. NMEAS is a local variable in FIGSEN defined when FIGSEN is called with NENTRY = 1. If the noise components associated with subsequent measurements are uncorrelated the standard deviation of the measurement is reduced $(NMEAS)^{\frac{1}{2}}$ times by the averaging process.

3.7 Actuator Calibration

The scale factors associated with the I' th mirror figure actuator is determined by perturbing the actuator input UFV(I) by an amount DACT and observing the actuator output UFAV(I) measured by a sensor. The scale DUMV(I) is calculated by the expression

$$DUMV(I) = \frac{(UFAV(I))_{UFV(I) = DACT} - (UFAV(I))_{UFV(I) = -DACT}}{2\,DACT} \qquad (3.7.1)$$

In order to minimize the effect of noise the measurements and scale computation are repeated NMEAS times and the results averaged. NMEAS is read in together with DACT. The subroutine prints the vector DUMV when all the scales factors have been determined.

3.8 Mirror Calibration

The reduced deformation - force matrix associated with the mirror is generated by perturbing the actuator outputs and measuring the resulting figure changes. The actuator command UFV(J) is modified and the resulting figure errors XFV(I) I = 1, N observed. The sensitivity coefficients are calculated from the equation

$$AM(I, J) = \frac{(XFV(I))_{UFV(J) = DACT} - (XFV(I))_{UFV(J) = -DACT}}{2\ DACT} \qquad (3.8\text{-}1)$$

where AM(I, J) is the I, J th element of the deformation force matrix AM.

In order to minimize the effects of figure sensor noise the test is repeated NMEAS times and the results averaged. NMEAS and DACT are read in by MIRCAL. If the noise components in subsequent measurements are uncorrelated the averaging process reduces the measurement deviation $(NMEAS)^{\frac{1}{2}}$ times.

CHAPTER 4

USER DESCRIPTIONS OF EAM SOFTWARE COMPONENTS

4.0 Introduction

The following pages contain user descriptions of the software components of the EAM. The descriptions list:

1. Program name.
2. The purpose of the program.
3. The calling statement form.
4. Definitions of the program parameters.
5. Card input data description.

The descriptions are divided into 4 sections:

1. EAM Package (EAM).
2. Parameter Optimization Package (POP).
3. Mathematical Operations Package (MOP).
4. Input Output Operations Package (IOP).

Listings of the programs in each section are contained in appendices A, B, C and D respectively.

The notation RBR in the data description indicates that the data is for a doubly dimensional array and is read in row by row.

Descriptions are provided for the following subprograms:

EXPERIMENTAL ACTIVE MIRROR SOFTWARE COMPONENTS

VARIABLE LIST:

EXPERIMENTAL ACTIVE MIRROR PACKAGE

AIM	INVERSE OF AM, TEMPORARY STORAGE
AM	DEFORMATION-FORCE MATRIX, TEMPORARY STORAGE
ARM	REDUCED VERSION OF AM
ARRM	REDUCED VERSION OF ARM
ASCALV	ACTUATOR COMMAND SCALE FACTORS
DA	TEMPORARY STORAGE FLOATING POINT
DTITCS	ACTUATOR SET TIME FOR ITERATIVE CONTROL SYSTEM
DUMV	TEMPORARY STORAGE VECTOR
FSCALV	FIGURE SENSOR MEASUREMENT SCALE FACTORS
GAINM	CONTROL SYSTEM GAIN MATRIX
GAINV	CONTROL SYSTEM GAIN VECTOR
I	INDEX
IA	TEMPORARY STORAGE INTEGER
ICHNG	SWITCH FOR MODIFYING VARIABLE VALUE
J	INDEX
K	INDEX
L	INDEX
LACTV	ACTUATOR LOCATION DESIGNATOR
MODE	TYPEWRITER CONTROL MODE
MODES	DESIRED VALUE OF MODE
MODOP	DESIRED CONTROL SYSTEM OPERATING MODE
MODOPS	DESIRED VALUE OF MODOP
N	NUMBER OF FIGURE MEASUREMENTS
NR	NUMBER OF ACTUATORS
NRA	DIMENSION OF GAINM
NSNSWT	NUMBER OF SENSE SWITCH USED TO INTERROGATE TYPEWRITER
TREAL	REAL TIME VARIABLE
UFAV	MEASURED ACTUATOR OUTPUTS
UFV	ACTUATOR INPUTS
XFRV	REDUCED XFV VECTOR
XFSV	X COORDINATES OF MEASUREMENT POINTS
XFV	MEASURED FIGURE ERRORS
XV	MAIN DATA BASE
YFSV	Y COORDINATES OF MEASUREMENT POINTS

PARAMETER OPTIMIZATION PACKAGE

EPS	STEP SIZE
GRADV	GRADIENT VECTOR
NHC	NUMBER OF HALVINGS COMPLETED
NHM	MAXIMUM NUMBER OF STEP SIZE HALVINGS
NIC	NUMBER OF ITERATIONS COMPLETED
NIM	MAXIMUM NUMBER OF SUCCESSFUL ITERATIONS
NITPRT	PRINT OUTPUT EVERY NITPRT ITERATIONS
NPAR	NUMBER OF ADJUSTABLE PARAMETERS
PARV	ADJUSTABLE PARAMETER VECTOR
PINDEX	PERFORMANCE INDEX VALUE
PISTOR	PERFORMANCE INDEX AT END OF LAST SUCCESSFUL ITERATION
SLGV	STORED VALUES OF GRADIENT VECTOR LENGTH
SPIV	STORED VALUES OF PERFORMANCE INDEX

NAME MAIN

PURPOSE TO INITIALIZE AND START ACTIVE MIRROR, TO ENGAGE REALTIME OPERATION MONITOR, AND TO GENERATE ACTUATOR COMMANDS

SUBROUTINE REQUIRED
- INIT
- IRANDP
- MFCS
- MODCHK
- RANDP
- REALT
- TYPCON

INPUT DATA
- NSNSWT
- (I10)

NAME ACTCAL

PURPOSE TO TEST FIGURE ACTUATORS

USAGE CALL ACTCAL(NENTRY)

PARAMETERS
- NENTRY=1 READ IN INITIALIZATION DATA
- NENTRY=2 ACTUATOR CALIBRATION

SUBROUTINES REQUIRED
- ACTCMD
- INIT
- IRANDP
- MXRNP
- RANDPD

INPUT DATA
- NMEAS
- (I10)
- DACT
- (E10.0)

NAME	ACTCMD	
PURPOSE	TO SCALE AND TRANSFER ACTUATOR COMMANDS AND ACTUATOR OUTPUT MEASUREMENTS.	
USAGE	CALL ACTCMD(NENTRY,I,UFV,UFAV,ASCALV)	
PARAMETERS	NENTRY=1	INITIALIZATION
	NENTRY=2	SCALE AND TRANSFER ACTUATOR COMMANDS
	I	DENOTES THE ITH ELEMENT OF UFV AND ASCALV TO BE USED
	UFV	ACTUATOR INPUT VECTOR
	UFAV	MEASURED ACTUATOR OUTPUT VECTOR
	ASCALV	ACTUATOR COMMAND SCALE FACTOR VECTOR

NAME	CHNG	
PURPOSE	TO SET X EQUAL TO V IF ICHNG IS EQUAL TO 1	
USAGE	CALL CHNG(ICHNG,V,X)	
PARAMETERS	ICHNG	IF ICHNG =1, SET X= V
	V	REAL VARIABLE
	X	REAL VARIABLE

NAME FIGSEN

PURPOSE TO MEASURE THE FIGURE ERROR AT A DISCRETE POINT ON THE REFLECTING SURFACE OF THE MIRROR

USAGE CALL FIGSEN(NENTRY,I,XFV,XFSV,YFSV,FSCALV)

PARAMETERS

NENTRY=1	INITIALIZATION
NENTRY=2	INTERROGATES FIGURE SENSOR AND SCALES MEASUREMENTS
I	DENOTES THE ITH ELEMENT OF XFSV AND YFSV TO BE USED
XFV	THE FIGURE ERROR VECTOR
XFSV	THE X COORDINATE VECTOR
YFSV	THE Y COORDINATE VECTOR
FSCALV	THE FIGURE ERROR SCALING VECTOR

SUBROUTINES REQUIRED
IRANDP
RANDPD

INPUT DATA
NMEAS
(I10)
PSCALE
(E10.0)

NAME INIT

PURPOSE TO INITIALIZE THE EXPERIMENTAL ACTIVE MIRROR

USAGE CALL INIT(NENTRY)

PARAMETERS

NENTRY=1	TO READ IN INITIALIZATION DATA
NENTRY=2	INITIALIZE MIRROR FIGURE CONTROL SYSTEM

SUBROUTINES REQUIRED
ACTCMD
FIGSEN
IMXRNP
IRANDP
MXRNP
TYPCON

INPUT DATA
N NR
(2I10)
AM
(RBR 7E10.0)
FSCALV
(7E10.0)
XFSV
(7E10.0)
YFSV
(7E10.0)
LACTV
(7I10)
ASCALV
(7E10.0)

NAME MFCS

PURPOSE TO REALIZE MIRROR FIGURE CONTROL SYSTEMS

USAGE CALL MFCS(NENTRY)

PARAMETERS

NENTRY=1	MIRROR FIGURE CONTROL SYSTEM INITIALIZATION
NENTRY=2	MEASURE FIGURE ERROR
NENTRY=3	ACTUATOR OUTPUT CONTROL SYSTEM
NENTRY=4	ITCS ACTUATOR COMMANDS

SUBROUTINES REQUIRED

ACTCMD
FIGSEN
MINFCN
MPRD
MTRA
MXRNP
RANDPD
REALT
REDUAM
SINV

INPUT DATA
GAINV
(7E10.0)
DTITCS
(E10.0)

NAME MIRCAL

PURPOSE TO CALIBRATE MIRROR

USAGE CALL MIRCAL(NENTRY)

PARAMETERS

NENTRY=1	READ IN INITIALIZATION DATA
NENTRY=2	MIRROR CALIBRATION

SUBROUTINES REQUIRED

ACTCMD
FIGSEN
INIT
IRANDP
MXRNP
RANDPD

INPUT DATA
NMEAS
(I10)
DACT
(E10.0)

NAME MODCHK

PURPOSE TO CHECK VALUES OF MODE AND MODOP

USAGE CALL MODCHK(NENTRY,MODE,MODED,MODOP,MODOPD,ITEST)

PARAMETERS NENTRY

MODE	TYPEWRITER CONTROL MODE
MODED	DESIRED VALUE OF MODE
MODOP	DESIRED CONTROL SYSTEM OPERATING MODE
MODOPD	DESIRED VALUE OF MODOP
ITEST	IS SET TO 2 IF MODE DOES NOT = MODED, OR IF MODOP DOES NOT = MODOPD

NAME REALT

PURPOSE TO INTERROGATE REAL TIME CLOCK

USAGE CALL REALT(TREAL)

PARAMETERS TREAL SUBROUTINE SETS TREAT TO THE REAL TIME

NAME REDUAM

PURPOSE TO GENERATE AR AND ARR FROM A

USAGE CALL REDUAM(NENTRY)

PARAMETERS

NENTRY=1	GENERATE AR BY REMOVING COLUMNS FROM A
NENTRY=2	GENERATE ARR FROM AR BY REMOVING ROWS FROM AR

SUBROUTINES REQUIRED LOC
MCPY
MXRNP

NAME TYPCON

PURPOSE TYPREWRITER CONTROL OF THE EXPERIMENTAL ACTIVE MIRROR

USAGE CALL TYPCON(NENTRY)

PARAMETERS NENTRY=1 READ IN INITIALIZATION DATA

NENTRY=2 TYPEWRITER CONTROL

SUBROUTINES REQUIRED ACTCAL
CHNG
ELMA
IRANDP
MIRCAL
REALT

PARAMETER OPTIMIZATION PACKAGE

NAME EPCHNG

PURPOSE TO LIMIT THE ABSOLUTE VALUE OF THE CHANGE IN THE COMPONENTS OF THE PARAMETER VECTOR.

USAGE CALL EPCHNG(NENTRY,STEP,STPDEC,DJDPV)

PARAMETERS NENTRY=1 INITIALIZE DATA

NENTRY=2 SELECT EPS BASED ON THE ABSOLUTE VALUE OF CHANGE IN THE PARAMETER VECTOR.

STEP STEP SIZE

STPDEC STEP DECREMENT VALUE

DJDPV GRADIENT VECTOR

SUBROUTINE REQUIRED MXRNP

INPUT DATA
DPARLV
(7E10.0)
PARMIN
(7E10.0)

NAME GRAD

PURPOSE TO CALCULATE GRADIENT VECTOR

USAGE CALL GRAD(NENTRY)

PARAMETERS

	NENTRY=1	INITIALIZE DATA
	NENTRY=2	CALCULATE GRADIENT

SUBROUTINES REQUIRED

ANGRAD
IRANDP
PINDX
RANDPD

INPUT DATA

DPAR
(E10.0)
NGRAD
(I10)

NAME ITPRT

PURPOSE TO PRINT OPTIMIZATION STATUS

USAGE CALL ITPRT(NENTRY)

PARAMETERS

	NENTRY=1	ITERATION OUTPUT
	NENTRY=2	RETURN
	NENTRY=3	RETURN
	NENTRY=4	RETURN
	NENTRY=5	ITPRT INITIALIZATION

SUBROUTINES REQUIRED

DESX
IRANDP
MXRNP
PINDX

INPUT DATA

NITPRT
(I10)

NAME MINFCN

PURPOSE TO MINIMIZE A PERFORMANCE INDEX

USAGE CALL MINFCN(NENTRY)

PARAMETERS NENTRY=1 INITIALIZE DATA

NENTRY=2 INITIALIZE PERFORMANCE INDEX

NENTRY=3 PERFORM MINIMIZATION

SUBROUTINES REQUIRED AVGRAD
CNGRAD
DAVIDN
EPCHNG
GRAD
GRADMX
IRANDP
ITPRT
MINFA
MXRNP
NEWRAF
PINDX
POWEL
RANDP
RANDPD

INPUT DATA

EPS EPSINC EPSDEC
(3E10.0)

NIM NHM NOPT NPAR ISDEC NSDEC
(6I10)

NPAR
(I10)

PARV
(7E10.0)

NAME PINDX

PURPOSE TO EVALUATE FIGURE PERFORMANCE INDEX

USAGE CALL PINDX(NENTRY)

PARAMETERS NENTRY=1 INITIALIZATION

NENTRY=2 CALCULATE PINDEX

SUBROUTINES REQUIRED FIGSEN

NAME	ELM	
PURPOSE	TO RETURN THE VALUE OF THE L,MTH ELEMENT OF THE MATRIX A	
USAGE	X= ELM(A,L,M,N)	
PARAMETERS	A	A MATRIX WITH N COLUMNS AND AN ARBITRARY NUMBER OF ROWS.
	L	THE ROW NUMBER OF THE ELEMENT BEING RETURNED
	M	THE COLUMN NUMBER OF THE ELEMENT BEING RETURNED
	N	THE NUMBER OF COLUMNS IN MATRIX A

NAME	ELMA	
PURPOSE	TO WRITE INTO AND READ FROM MEMORY THE I,JTH ELEMENT OF MATRIX A	
USAGE	CALL ELMA(NENTRY,A,I,J,V,N)	
PARAMETERS	NENTRY=1	THE VALUE OF V IS STORED INTO THE I,JTH ELEMENT OF A
	NENTRY=2	THE VALUE OF THE I,JTH ELEMENT OF A IS STORED INTO V
	A	THE MATRIX WHOSE I,JTH ELEMENT IS USED
	I	THE ROW OF THE ELEMENT IN MATRIX A
	J	THE COLUMN OF THE ELEMENT IN MATRIX A
	V	A SCALAR QUANTITY
	N	THE NUMBER OF ROWS IN A

NAME	GMADD	
PURPOSE	TO PERFORM MATRIX ADDITION, R=A+B	
USAGE	CALL GMADD(A,B,R,N,M)	
PARAMETERS	A	AN N BY M MATRIX
	B	AN N BY M MATRIX
	R	AN N BY M MATRIX
	N	THE NUMBER OF ROWS IN A,B, AND R
	M	THE NUMBER OF COLUMNS IN A,B, AND R

NAME	GMPRD	
PURPOSE	TO FORM THE PRODUCT, R=A*B	
USAGE	CALL GMPRD(A,B,R,N,M,L)	
PARAMETERS	A	AN N BY M MATRIX
	B	AN M BY L MATRIX
	R	AN N BY L MATRIX
	N	THE NUMBER OF ROWS IN A AND IN R
	M	THE NUMBER OF COLUMNS IN A ; ALSO THE NUMBER OF ROWS IN B.
	L	THE NUMBER OF COLUMNS IN B; ALSO, THE NUMBER OF COLUMNS IN R.

NAME	GMSUB	
PURPOSE	TO PERFORM MATRIX SUBTRACTION, R=A-B	
USAGE	CALL GMSUB(A,B,R,N,M)	
PARAMETERS	A	AN N BY M MATRIX
	B	AN N BY M MATRIX
	R	AN N BY M MATRIX
	N	THE NUMBER OF ROWS IN A,B, AND R
	M	THE NUMBER OF COLUMNS IN A,B, AND R

NAME	GMTRA	
PURPOSE	TO TRANSPOSE MATRIX A INTO MATRIX R	
USAGE	CALL GMTRA(A,R,N,M)	
PARAMETERS	A	AN N BY M MATRIX
	R	AN M BY N MATRIX
	N	THE NUMBER OF ROWS IN A; ALSO, THE NUMBER OF COLUMNS IN R.
	M	THE NUMBER OF COLUMNS IN A; ALSO, THE NUMBER OF ROWS IN R.

NAME GTOSYM

PURPOSE TO CONVERT A SQUARE SYMMETRIC NX BY NX MATRIX INTO A SYMMETRIC MATRIX IN SUPPRESSED SYMMETRIC STORAGE. THE MATRIX PRODUCED CONSISTS OF THE UPPER TRIANGLE (INCLUDING THE DIAGONAL ELEMENTS) OF THE NX BY NX MATRIX. THE LENGTH OF THE VECTOR CREATED IS NX*(NX+1)/2.

USAGE CALL GTOSYM(X,XS,NX)

PARAMETERS

X	THE NX BY NX SYMMETRIC MATRIX (INPUT)
XS	THE UPPER TRIANGLE FORM PRODUCED (OUTPUT)
NX	THE ORDER OF THE INPUT MATRIX

NAME LOC

PURPOSE TO COMPUTE A VECTOR SUBSCRIPT FOR AN ELEMENT IN A MATRIX OF SPECIFIED STORAGE MODE.

USAGE CALL LOC(I,J,IR,N,M,MS)

PARAMETERS

I	ROW NUMBER OF ELEMENT
J	COLUMN NUMBER OF ELEMENT
IR	RESULTANT VECTOR SUBSCRIPT (DETERMINED BY LOC)
N	NUMBER OF ROWS IN MATRIX
M	NUMBER OF COLUMNS IN MATRIX
MS	A SINGLE DIGIT INDICATING THE STORAGE MODE OF THE MATRIX. 0 GENERAL 1 SYMMETRIC 2 DIAGONAL

NAME MCPY

PURPOSE TO COPY MATRIX A INTO MATRIX R

USAGE CALL MCPY(A,R,N,M,MS)

PARAMETERS

A	AN N BY M MATRIX
R	AN N BY M MATRIX
N	THE NUMBER OF ROWS IN A AND IN R
M	THE NUMBER OF COLUMNS IN A AND IN R
MS	A SINGLE DIGIT INDICATING STORAGE MODE OF BOTH A AND R 0 GENERAL 1 SYMMETRIC 2 DIAGONAL

SUBROUTINES REQUIRED LOC

NAME MMADD

PURPOSE TO FORM THE COMBINATION, C=ALPHA*A+BETA*B

USAGE CALL MMADD(N,ALPHA,BETA,B,C)

PARAMETERS

N	LENGTH OF VECTORS A,B, AND C
ALPHA	SCALAR QUANTITY
A	VECTOR OF LENGTH N
BETA	SCALAR QUANTITY
B	VECTOR OF LENGTH N
C	VECTOR OF LENGTH N

NAME MPRD

PURPOSE TO FORM THE PRODUCT, R=A*B

USAGE CALL MPRD(A,B,R,N,M,MSA,MSB,L)

PARAMETERS

A	AN N BY M MATRIX
B	AN M BY L MATRIX
R	AN N BY L MATRIX
N	THE NUMBER OF ROWS IN A AND IN R
M	THE NUMBER OF COLUMNS IN A; ALSO, THE NUMBER OF ROWS IN B.
MSA	A SINGLE DIGIT INDICATING STORAGE MODE FOR MATRIX A 0 GENERAL 1 SYMMETRIC 2 DIAGONAL
MSB	A SINGLE DIGIT INDICATING STORAGE MODE FOR MATRIX B 0 GENERAL 1 SYMMETRIC 2 DIAGONAL
L	THE NUMBER OF COLUMNS IN B; ALSO, THE NUMBER OF COLUMNS IN R.

SUBROUTINES REQUIRED LOC

NAME MTRA

PURPOSE TO TRANSPOSE MATRIX A TO FORM MATRIX R

USAGE CALL MTRA(A,R,N,M,MS)

PARAMETERS

A	AN N BY M MATRIX
R	AN M BY N MATRIX
N	THE NUMBER OF ROWS IN A; ALSO, THE NUMBER OF COLUMNS IN R.
M	THE NUMBER OF COLUMNS IN A; ALSO, THE NUMBER OF ROWS IN R.
MS	A SINGLE DIGIT INDICATING THE STORAGE MODE OF BOTH A AND R. 0 GENERAL 1 SYMMETRIC 2 DIAGONAL

SUBROUTINES REQUIRED MCPY

NAME SYMTOG

PURPOSE TO CONVERT A SYMMETRIC MATRIX (IN SUPPRESSED SYMMETRIC STORAGE), WHOSE LENGTH IS NX*(NX+1)/2, INTO A SQUARE SYMMETRIC NX BY NX MATRIX WHOSE LENGTH IS NX*NX.

USAGE CALL SYMTOG(XS,X,NX)

PARAMETERS

XS	THE SYMMETRIC MATRIX VECTOR (INPUT)
X	THE EXPANDED GENERAL MATRIX VECTOR (OUTPUT)
NX	THE ORDER OF THE INPUT MATRIX

INPUT OUTPUT OPERATIONS PACKAGE

NAME IMXRNP

PURPOSE TO READ AND PRINT INTEGER-VALUED MATRICES

USAGE CALL IMXRNP(MA,NA,NB,NENTRY)

PARAMETERS

MA	1-DIMENSION INTEGER VECTOR WHOSE DIMENSION MUST BE AT LEAST NA*NB. MATRIX IS STORED COLUMN-WISE IN THIS VECTOR.
NA	NUMBER OF ROWS IN MATRIX
NB	NUMBER OF COLUMNS IN MATRIX
NENTRY=1	READ IN AND PRINT OUT MATRIX
NENTRY=2	READ IN MATRIX
NENTRY=3	PRINT OUT MATRIX
NENTRY=4	READ IN HEADING CARD, READ IN MATRIX AND PRINT OUT MATRIX.
NENTRY=5	PUNCH OUT MATRIX

SUBROUTINES REQUIRED RANDP

INPUT DATA INTEGER MATRIX (ROW-WISE)

OUTPUT PRINTED INTEGER MATRIX (ROW-WISE)

NAME IRANDP

PURPOSE TO READ AND PRINT INTEGER DATA

USAGE CALL IRANDP(ND,IA,IB,IC,ID,IE,IF,IG,NENTRY)

PARAMETERS

ND	THE NUMBER OF INTEGER VALUES TO BE READ IN AND PRINTED OUT.
IA-IG	THE INTEGER VARIABLES TO WHICH THE INTEGER VALUES WILL BE ASSIGNED.
NENTRY=1	READ IN 7 INTEGER VALUES (7I10) FOR THE INTEGER VARIABLES IA-IG AND PRINT THE FIRST ND VARIABLES.
NENTRY=2	PRINT THE FIRST ND INTEGER VARIABLES
NENTRY=3	RETURN
NENTRY=4	READ AND PRINT HEADING CARD, READ 7 INTEGER VALUES THE FIRST ND VARIABLES. (7I10) FOR THE INTEGER VARIABLES IA-IG AND PRINT OUT

SUBROUTINES REQUIRED RANDP

INPUT DATA MAY BE HEADING CARD
INTEGER VALUES CARD

OUTPUT MAY BE PRINTED HEADING
PRINTED INTEGER VALUES

NAME MXRNP

PURPOSE TO READ AND PRINT REAL-VALUED MATRICES

USAGE CALL MXRNP(VA,NA,NB,NENTRY)

PARAMETERS

VA	1-DIMENSION REAL VECTOR WHOSE DIMENSION MUST BE AT LEAST NA*NB. MATRIX IS STORED COLUMN-WISE IN THIS VECTOR.
NA	NUMBER OF ROWS IN MATRIX
NB	NUMBER OF COLUMNS IN MATRIX
NENTRY=1	READ IN AND PRINT OUT MATRIX
NENTRY=2	READ IN MATRIX
NENTRY=3	PRINT OUT MATRIX
NENTRY=4	READ IN HEADING CARD, READ IN MATRIX AND PRINT OUT MATRIX.
NENTRY=5	PUNCH OUT MATRIX

SUBROUTINES REQUIRED RANDP

INPUT DATA REAL MATRIX (ROW-WISE)

OUTPUT PRINTED REAL MATRIX (ROW-WISE)

NAME	NAMRNP	
PURPOSE	TO READ,PRINT AND STORE MATRIX M WHICH CONTAINS 4-CHARACTER NAMES	
USAGE	CALL NAMRNP(M,NA,NB,NENTRY)	
PARAMETERS	M	AN NA BY NB MATRIX
	NA	THE NUMBER OF ROWS IN M
	NB	THE NUMBER OF COLUMNS IN M
	NENTRY=1	READ IN AND PRINT M AND STORED M INTO A SINGLE-DIMENSIONED VECTOR.
	NENTRY=2	READ IN MATRIX M AND STORE M INTO A SINGLE-DIMENSIONED VECTOR.
	NENTRY=3	PRINT OUT MATRIX M
	NENTRY=4	READ IN A HEADING CARD, READ IN MATRIX M AND STORE M INTO A SINGLE-DIMENSIONED VECTOR, AND PRINT M
SUBROUTINES REQUIRED		RANDP
INPUT DATA	AN NA BY NB MATRIX OF 4-CHARACTER NAMES	
OUTPUT	PRINT THE NA BY NB MATRIX	

NAME	RANDP	
PURPOSE	TO READ AND PRINT HEADING CARDS	
USAGE	CALL RANDP(NENTRY)	
PARAMETERS	NENTRY=1	READS CARD IN 8A8 FORMAT AND PRINTS CONTENTS (AT TOP OF NEXT PAGE) IN 8A8 FORMAT.
	NENTRY=2	READS CARD IN 8A8 FORMAT AND PRINTS CONTENTS IN 8A8 FORMAT.
	NENTRY=3	READS CARD IN 7(2X,A8) FORMAT AND PRINTS CONTENTS IN 7(7X,A8) FORMAT AT TOP OF NEXT PAGE.
	NENTRY=4	READS CARD IN 7(2X,A8) FORMAT AND PRINTS CONTENTS IN 7(7X,A8) FORMAT.
INPUT DATA	HEADING CARD	
OUTPUT	PRINTED HEADING	

NAME RANDPD

PURPOSE TO READ AND PRINT FLOATING POINT DATA

USAGE CALL RANDPD(ND,DA,DB,DC,DE,DF,DG,NENTRY)

PARAMETERS

ND	THE NUMBER OF REAL FLOATING PT. VALUES TO BE READ IN AND/OR PRINTED OUT.
DA-DG	THE REAL VARIABLES TO WHICH THE REAL VALUES WILL BE ASSIGNED.
NENTRY=1	READ IN 7 REAL VALUES (7E10.0) FOR REAL VARIABLES DA-DG AND PRINT THE FIRST ND VARIABLES.
NENTRY=2	PRINT OUT THE FIRST ND VARIABLES.
NENTRY=3	READ IN REAL VALUES (7E10.0) FOR REAL VARIABLES DA-DG
NENTRY=4	READ AND PRINT HEADING CARD, READ IN 7 REAL VALUES (7E10.0) FOR THE REAL VARIABLES DA-DG, AND PRINT OUT THE FIRST ND VARIABLES.

SUBROUTINES REQUIRED RANDP

INPUT DATA MAY BE HEADING CARD
FLOATING PT. VALUES CARD

OUTPUT MAY BE PRINTED HEADING
PRINTED FLOATING PT. VALUES

PRECEDING PAGE BLANK NOT FILMED

CHAPTER 5

TYPICAL DATA CONFIGURATIONS

5.0 Introduction

Data for the EAM is introduced in two different fashions: card input and typewriter input. Typewriter input data is detailed in section 3. 2. Card input data comprises the bulk of the information utilized by the EAM software. The following sections discuss the card data deck structure

5.1 Card Input Data for the SLCS and LOCS

If Simplified Linear (SLCS) or Linear Optimal (LOCS) Control is desired, the following data deck is read by the card reader:

NSNSWT	
(2I10)	
N	NR
(2I10)	
AM	
(RBR 7E10. 0)	
FSCALV	
(7E10. 0)	
XFSV	
(7E10. 0)	
YFSV	
(7E10. 0)	
LACTV	
(7E10. 0)	
ASCALV	
(7E10. 0)	
NMEAS	
(I10)	
PSCALE	
(E10. 0)	
NMEAS	
(I10)	
DACT	
(E10. 0)	
NMEAS	
(I10)	
DACT	
(E10. 0)	
GAINV	
(7E10. 0)	
DTITCS	
(E10. 0)	

The data deck includes variable heading cards, to assist identification, as well as the numerical data in the indicated format. The heading card is read and printed by the computer over the corresponding data in the output print. The notation (RBR 7E10.0) under the heading AM indicates that the variable AM (the force deformation matrix of the mirror) is read in row by row in 7E10.0 (unassigned decimal) format. Each row must be started on a new card.

5.2 Card Input Data for ITCS

If the iterative figure control system (ITCS) is to be realized, the software will read in the data deck described in section 5.1, and the following set of heading and numerical data cards:

EPS (3E10.0)	EPSINC	EPSDEC				
NIM (6I10)	NHM	NOPT	NPAR	ISDEC	NSDEC	
NPAR (I10)						
PARV (7E10.0)						
DPAR (E10.0)						
NGRAD (I10)						
DPARLV (7E10.0)						
PARMIN (E10.0)						
NITPRT (I10)						

CHAPTER 6

SPECIFICATION OF A SMALL HYBRID COMPUTER FOR THE EAM

6.0 Introduction

A key part of the study of advanced space telescope technology currently in progress at NASA-MSFC and MIT-DL will be an experimental program which will lead to the evaluation and development of techniques for actively controlling the optical surface shape or figure of primary telescope mirrors. A necessary feature of the experimental program is a control system which will feature an element (or set of elements) which are easily and inexpensively modified to reflect control algorithm design changes.

Adopting a modern approach, it was decided to use a small general purpose digital computer as the programmable control system component. Such an approach is economical as a result of the significant reduction in small computer manufacturing costs, which have occurred over the last few years, and the high cost of constructing special purpose analog systems.

The digital computer also provides a valuable tool for real-time processing of experimental data and the solution of other control problems associated with system alignment.

The main disadvantage of the digital computer is the character of the information it handles. Each piece of information in the computer is characterized by a binary number. The number is expressed by one dimensional array of N elements each of which can have two values which are identified by 0 and 1. A power of 2 is associated with each element. Hardware devices in control systems, on the other hand,

generally produce data which is of a one dimensional, or analog, character. Thus conversion devices must be used which map the one dimensional analog signals into N dimensional binary numbers for computation purposes. Converters are also required to map N dimensional binary data into one dimensional signals for use by hardware elements. Both of these operations are easily performed using high-speed digital to analog and analog to digital converters.

After discussions with a number of people at MIT-DL who are using the Digital Equipment Corporation PDP-9 (a predecessor of the PDP-15) and Dean Hamilton (formerly in charge of the PDP-10 digital computer facility at NASA-ERC and currently director of the CARS computer at MIT-DL), it was decided to restrict attention to the Digital Equipment Corporation PDP-15 Computer.[12-13]

6.1 A Small Computer for the EAM

6.1.1 Manufacturer

Digital Equipment Corporation
146 Main Street
Maynard, Massachusetts 01754

6.1.2 Manufacturer's Designation

Model PDP-15

6.1.3 Basic Computer

The basic computer consists of a central processor and a random access core memory. The central processor is capable of executing a set of instructions which include fixed point arithmetic operations on data (add, subtract, multiply and divide); data transfer operations to and from the core memory, mass storage devices,* and peripherals such as the teleprinter and other data displays;** and logic operations such as test and branch. The basic core memory consists of 4,096 words. The memory cycle time*** is 0.8 microseconds which is relatively fast.

6.1.4 Memory Expansions

The random access core memory can be expanded in 4,096 word increments to a maximum of 32,768 words.****

---

* Tape units, disks

** Cathode ray plotter, calcomp plotter, high-speed printer

*** Memory "Cycle time" is defined as the time required to transfer a data word to or from the core memory.

**** Further expansion to 131,072 words is possible with hardware modifications.

6.1.5 Analog to Digital Conversion

The conversion is performed by one converter which is sequentially switched to each input channel. Each channel level is converted to a binary number which is transferred to the central processor. The analog signal switching is performed by field effect transistor switches. The converter features a selectable 6 to 12 bit output word length which provides more than enough resolution* for handling figure sensor data.

6.1.6 Digital to Analog Conversion

Digital to analog conversion is normally performed by transferring the binary number into a buffer register where it is stored.** The stored binary number is then observed by a device which produces an analog signal proportional to the value of the binary number. The resolution in the analog signal is determined by the number of bits in the buffer register. A length of at least 10 bits is required to yield acceptable resolution (1 part in 1,024).

---

* If the maximum figure error is 5λ and the minimum sensed error is $\lambda/200$, the ratio $(5\lambda)/(\lambda/200) = 1,000 < 2^{10} = 1,024$. The actuator commands are linearly related to the figure errors: thus, the same criteria may be applied to the output signals from the computer which drives the actuators.

** An operational amplifier with an appropriate network of input resistors.

6.1.7 Mass Memory

DEC manufactures a wide variety of tape transports and disc memories for the PDP-15. The small Dectape units are inexpensive and can virtually eliminate the necessity of using paper tape or cards for storage (a single reel of Dectape can store 150,000 18 bit words).

6.1.8 Data Communications

The PDP-15 is designed for use with a variety of teletypes, high-speed printers, paper-tape readers and punches, card readers and cathode ray plotting devices.

6.1.9 Special Options

A particularly useful control processor option is the extended arithmetic element which reduces the time required to perform multiplications and divisions by a factor of approximately 50. Since computer utilization in a real-time environment is usually limited by computation speed rather than memory capacity, reducing the time required to perform multiplications and divisions (the most time-consuming operations) can significantly increase the usefulness of the computer.

6.2 Component Cost for PDP-15

6.2.1 Basic System

Central Processor		
4,096 18 bit words of core memory	PDP-15	$15,600

6.2.2	Memory Expansion for Basic System		
	4,096 word core module to increase Basic System memory capacity to 8,192 words	MK15-A	6,000
6.2.3	Memory Modules for Further Expansion		
	Assembly to further increase memory capacity by 8,192 words	MM15-A/MK15-A	14,000
6.2.4	Console Teleprinter (Heavy Duty)	KSR-35	3,000
6.2.5	High-Speed Paper Tape Reader and Punch	PC 15	3,900
	Paper tape input is required to run diagnostic routines		
6.2.6	Analog to Digital Conversion		
	6 to 12 bit selectable analog to digital converter and multiplexer control	AF03B	5,000
	9-35 μsec conversion time multiplexes up to 64 channels	A121	65
	4-Channel field effect transistor switch: One required for each set of four analog to digital channels		
	Scaling amplifiers (if required)	AF01B	300
	Convert input signals to an average		

	level and amplitude suitable for the analog to digital converter and FET switches (one for each channel)		
6.2.7	Digital to Analog Conversion		
	Multiplexer control for up to 16, 10 bit, digital to analog converter channels	AA05A	$ 5,500
	Expansion of AA05A to 64 channels	AA05B	2,900
	Digital to analog converter, single buffered ± 5.0 volts or ± 10.0 volts: One required per channel	A609	375
	Reference power supply: For up to 12 digital to analog converters	A610 or A611	400
6.2.8	DEC Tape Drives		
	DEC tape control unit for 8 tape units	TC02D	5,400
	Dectape transport	TU55	2,350
6.2.9	Extended Arithmetic Element	KE15	2,800
	Decreases multiply and divide time by approximately 50		
6.2.10	Positive to Negative Buss Converter	DW15A	2,000
	Required for compatibility by some D/A and A/D modules		
6.2.11	Disk Memory	RP02	9,000

Stores up to 262,144 words per disk, average access time* 16.7 milliseconds		
Control unit for up to 8 RP02 disks	RP15	$6,000

6.3 Prices of Typical System Configurations

6.3.1 Basic System Expanded to 16,384 Words of Core Memory, Teleprinter, Paper-Tape Reader and Punch

1 (PDP-15/10)	$15,600
2 (MK15-A)	12,000
1 (MM15-A)	8,000
1 (KSR-35)	3,000
1 (PC 15)	3,900
TOTAL	$42,500

6.3.2 System 6.3.1 Expanded to Include Three DEC Tape Units

1 (6.3.1)	$42,500
1 (TC02D)	5,400
3 (TU55)	7,050
TOTAL	$54,950

* Time required to acquire a piece of information stored on the disk: Access time is primarily a function of disk angular velocity.

6.3.3 Basic Digital to Analog Package (16 Channels)

1 (AA05A)	$5,500
16 (A609)	6,000
2 (A610)	1,600
TOTAL	$13,100

6.3.4 Basic Analog to Digital Package (16 Channels)

1 (AF01B)	$ 5,000
4 (A121)	260
16 (AH03)	4,800
TOTAL	$10,060

6.3.5 System 4.2 with D/A and A/D Packages 4.3 and 4.4

1 (3.2)	$54,950
1 (3.3)	13,100
1 (3.4)	10,060
TOTAL	$78,110

6.4 Summary

The system defined in Section 3.5 represents a minimal system for an active mirror experimental program. Such a hybrid computer would permit configurations using up to 16 actuators and sensor outputs to be investigated. More complex systems will probably require an increase

in core memory and modifications in the A/D and D/A channeling to carry the increased load.

CHAPTER 7
SUMMARY AND CONCLUSIONS

The software requirements for an experimental active mirror have been defined and a complete preliminary software package design has evolved. Although the software was specifically designed for compatibility with the XDS Sigma 5/7 system, its FORTRAN II-IV structure permits it to be executed by practically any FORTRAN operating system with minor modifications.

The software package realizes the linear optimal, simplified linear and iterative control algorithms discussed in reference 1. In addition the software provides servo loop closures about the figure actuators which help to eliminate the effect of actuator characteristics variations.

The software package includes routines for actuator calibration and the generation of discrete mirror models. An operating package is also incorporated which enables the investigator to control the operation of the EAM and to operate the system in diagnostic or system parameter modification modes.

It should be emphasized that the software has not been thoroughly debugged. Rigorous testing will be an important segment of the follow-on work associated with this project. Testing will include the definition of software models of the hardware components so that the software package can be completely debugged without the necessity of communicating with hardware elements. This will enable most of the software tests to be performed using the MIT computer facility.

The report concludes with a description of a small hybrid computer (DEC PDP 15) for application in an experimental active mirror. While the specified computer facility is not directly applicable to the EAM as it is currently envisioned it is useful to review the technical considerations involved in the selection of the PDP 15 and its peripherals.

APPENDIX A

EAM PACKAGE

A.0 Introduction

This appendix contains listings of the following subroutines.

```
C      MAIN                                                                    EAM10000
C                                                                              EAM10010
C      MAIN PROGRAM FOR EXPERIMENTAL ACTIVE MIRROR                             EAM10020
C                                                                              EAM10030
       DIMENSION XV(10000),AM(3000),AIM(3000),XFV(20),UFV(20),ASCALV(20),      EAM10040
      1 FSCALV(20),XFSV(20),YFSV(20),XFRV(20),DUMV(20),UFAV(20),DUMVA(20)      EAM10050
      2 ,GAINV(10),GAINM(400),LACTV(20)                                        EAM10060
       COMMON/BLKMFC/ N,NR,XV,MODE,LACTV,MODOP,NSNSWT,NRA                      EAM10070
       EQUIVALENCE (XV(1),AM(1)),(XV(3001),AIM(1)),(XV(6001),XFV(1)),          EAM10080
      1 (UFV(1),XV(6101)),(ASCALV(1),XV(6121)),(FSCALV(1),XV(6141)),           EAM10090
      2 (XFSV(1),XV(6161)),(YFSV(1),XV(6181)),(XFRV(1),XV(6201)),              EAM10100
      3 (UFAV(1),XV(6221)),(DUMV(1),XV(6241)),(DUMVA(1),XV(6261)),             EAM10110
      4 (GAINV(1),XV(6281)),(GAINM(1)(XV(6291))                                EAM10120
C                                                                              EAM10130
 1000 FORMAT(1H1,/,4HMAIN)                                                     EAM10140
 1010 FORMAT(9HMODOP ERR)                                                      EAM10150
C                                                                              EAM10160
C      INITIALIZATION                                                          EAM10170
       PRINT 1000                                                             EAM10180
       CALL RANDP(2)                                                          EAM10190
       CALL IRANDP(1,NSNSWT,IA,IA,IA,IA,IA,IA,4)                              EAM10200
       CALL TYPCON(1)                                                         EAM10210
       CALL INIT(1)                                                           EAM10220
 2105  CALL TYPCON(2)                                                         EAM10230
C                                                                              EAM10240
C      IS CONTROL CONFIGURATION DEFINED                                        EAM10250
       IF(MODOP-5)2110,2110,2120                                              EAM10260
C      NO                                                                      EAM10270
 2110  TYPE 1010                                                              EAM10280
       GO TO 2105                                                             EAM10290
C      YES                                                                     EAM10300
 2120  MODOPS=MODOP                                                           EAM10310
C                                                                              EAM10320
C      INITIALIZE ACTIVE MIRROR                                                EAM10330
       CALL TYPCON(2)                                                         EAM10340
       MODES=1                                                                EAM10350
       CALL MODCHK(1,MODE,MODES,MODOP,MODOPS,IA)                              EAM10360
       GO TO (2130,2105),IA                                                   EAM10370
 2130  CALL INIT(2)                                                           EAM10380
       CALL MFCS(1)                                                           EAM10390
       IF (MODOP-8) 2133, 2132, 2133                                          EAM10391
 2132  CALL MINFCN (1)                                                        EAM10392
       CALL MINFCN (2)                                                        EAM10393
C                                                                              EAM10400
C      START ACTIVE MIRROR                                                     EAM10410
 2133  MODES=2                                                                EAM10420
 2135  CALL TYPCON(2)                                                         EAM10430
       CALL MODCHK(1,MODE,MODES,MODOP,MODOPS,IA)                              EAM10440
       GO TO (2140,2135),IA                                                   EAM10450
 2140  CALL REALT(TREAL)                                                      EAM10460
       TSTORE=TREAL                                                           EAM10470
       GO TO 2150                                                             EAM10480
C                                                                              EAM10490
C      REALTIME OPERATION MONITOR                                              EAM10500
 2141  XV(6151)=0.0                                                          EAM10510
 2142  CALL REALT(TREAL)                                                      EAM10520
       IF(TREAL-TSTORE-DT)2142,2143,2143                                      EAM10530
 2143  TSTORE=TREAL                                                           EAM10540
C                                                                              EAM10550
C                                                                              EAM10560
```

```
C      GENERATE ACTUATOR COMMANDS                                        EAM10570
 2150 CALL MFCS(2)                                                       EAM10580
C                                                                        EAM10590
C      INTERROGATE TYPEWRITER IF SENSE SWITCH NSNSWT IS SET              EAM10600
 2145 IF(SENSE SWITCH NSNSWT) 2205,2147                                  EAM10610
 2205 CALL REALT(TREAL)                                                  EAM10620
      DA=TREAL-TSTORE                                                    EAM10630
 2146 CALL TYPCON(2)                                                     EAM10640
      CALL MODCHK(1,MODE,MODES,MODOP,MODOPS,IA)                          EAM10650
      GO TO (2147,2146)IA                                                EAM10660
 2147 CALL REALT(TREAL)                                                  EAM10670
      TSTORE=TREAL-DA                                                    EAM10680
      GO TO 2141                                                         EAM10690
C                                                                        EAM10700
      END                                                                EAM10710
```

```
      SUBROUTINE ACTCAL(NENTRY)                                          EAM10720
C                                                                        EAM10730
C     SUBROUTINE TO TEST FIGURE ACTUATORS                                EAM10740
C                                                                        EAM10750
      DIMENSION XV(10000),AM(3000),AIM(3000),XFV(20),UFV(20),ASCALV(20),EAM10760
     1 FSCALV(20),XFSV(20),YFSV(20),XFRV(20),DUMV(20),UFAV(20),DUMVA(20)EAM10770
     2 ,GAINV(10),GAINM(400),LACTV(20)                                  EAM10780
      COMMON/BLKMFC/ N,NR,XV,MODE,LACTV,MODOP,NSNSWT,NRA                 EAM10790
      EQUIVALENCE (XV(1),AM(1)),(XV(3001),AIM(1)),(XV(6001),XFV(1)),     EAM10800
     1 (UFV(1),XV(6101)),(ASCALV(1),XV(6121)),(FSCALV(1),XV(6141)),      EAM10810
     2 (XFSV(1),XV(6161)),(YFSV(1),XV(6181)),(XFRV(1),XV(6201)),         EAM10820
     3 (UFAV(1),XV(6221)),(DUMV(1),XV(6241)),(DUMVA(1),XV(6261)),        EAM10830
     4 (GAINV(1),XV(6281)),(GAINM(1)(XV(6291))                          EAM10840
C                                                                        EAM10850
 1000 FORMAT(6HACTCAL)                                                   EAM10860
 1002 FORMAT(1H1/3X,12HACTOUT/ACTIN)                                     EAM10870
 1003 FORMAT(7X,8HACAL END)                                              EAM10880
C                                                                        EAM10890
      GO TO (1,2),NENTRY                                                 EAM10900
C                                                                        EAM10910
C     INPUT DATA                                                         EAM10920
 1    PRINT 1000                                                         EAM10930
      CALL IRANDP(1,NMEAS,IA,IA,IA,IA,IA,IA,4)                           EAM10940
      CALL RANDPD(1,DACT,DA,DA,DA,DA,DA,DA,4)                            EAM10950
      DB=DACT*2.0                                                        EAM10960
      RETURN                                                             EAM10970
C                                                                        EAM10980
C     ACTUATOR CALIBRATION                                               EAM10990
 2    CALL INIT(2)                                                       EAM11000
      DO 2202 I=1,NR                                                     EAM11010
      DO 2201 J=1,NMEAS                                                  EAM11020
      UFV(I)=-DACT                                                       EAM11030
      CALL ACTCMD(2,I,UFV,UFAV,ASCALV)                                   EAM11040
      DA=UFAV(I)                                                         EAM11050
      UFV(I)=DACT                                                        EAM11060
      CALL ACTCMD(2,I,UFV,UFAV,ASCALV)                                   EAM11070
 2201 DUMV(I)=(UFAV(I)-DA)/DB+DUMV(I)                                    EAM11080
 2202 DUMV(I)=DUMV(I)/NMEAS                                              EAM11090
C                                                                        EAM11100
C     PRINT OUT ACTUATOR SCALE VECTOR                                    EAM11110
      PRINT 1002                                                         EAM11120
      CALL MXRNP(DUMV,1,NR,3)                                            EAM11130
      TYPE 1003                                                          EAM11140
      RETURN                                                             EAM11150
C                                                                        EAM11160
      END                                                                EAM11170
```

```
      SUBROUTINE ACTCMD(NENTRY,I,UFV,UFAV,ASCALV)                       EAM11180
C                                                                       EAM11190
C     SUBROUTINE TO SCALE AND TRANSFER ACTUATOR COMMANDS AND ACTUATOR OUEAM11200
C     MEASUREMENTS.                                                     EAM11210
C                                                                       EAM11220
      DIMENSION UFV(1),UFAV(1),ASCALV(1)                                EAM11230
C                                                                       EAM11240
      GO TO(1,2),NENTRY                                                 EAM11250
C                                                                       EAM11260
C     INITIALIZATION                                                    EAM11270
 1    RETURN                                                            EAM11280
C                                                                       EAM11290
C     SCALE AND TRANSFER ACTUATOR COMMANDS                              EAM11300
 2    DA=UFV(I)*ASCALV(I)                                               EAM11310
C                                                                       EAM11320
C                                                                       EAM11330
C     INSERT DTOA SOFTWARE HERE                                         EAM11340
C                                                                       EAM11350
C                                                                       EAM11360
C     TRANSFER ACTUATOR OUTPUT MEASUREMENTS                             EAM11370
C                                                                       EAM11380
C                                                                       EAM11390
C     INSERT ATOD SOFTWARE HERE                                         EAM11400
C                                                                       EAM11410
C                                                                       EAM11420
      UFAV(I)=DA                                                        EAM11430
      RETURN                                                            EAM11440
C                                                                       EAM11450
      END                                                               EAM11460
```

```
      SUBROUTINE CHNG(ICHNG,V,X)                                        EAM11470
C                                                                       EAM11480
C     SUBROUTINE SETS X EQUAL TO V IF ICHNG IS EQUAL TO 1               EAM11490
C                                                                       EAM11500
      GO TO (1,2),ICHNG                                                 EAM11510
C                                                                       EAM11520
 1    X=V                                                               EAM11530
 2    RETURN                                                            EAM11540
C                                                                       EAM11550
      END                                                               EAM11560
```

```
      SUBROUTINE FIGSEN(NENTRY,I,XFV,XFSV,YFSV,FSCALV)                  EAM11570
C                                                                       EAM11580
C     SUBROUTINE TO MEASURE THE FIGURE ERROR XFV(I) AT A DISCRETE POINT EAM11590
C     COORDINATES XFSV(I),YFSV(I) ON THE REFLECTING SURFACE OF THE MIRROEAM11600
C                                                                       EAM11610
      DIMENSION XFV(1),XFSV(1),YFSV(1)                                  EAM11620
C                                                                       EAM11630
 1000 FORMAT(6HFIGSEN)                                                  EAM11640
C                                                                       EAM11650
      GO TO(1,2),NENTRY                                                 EAM11660
C                                                                       EAM11670
C     INITIALIZATION                                                    EAM11680
  1   PRINT 1000                                                        EAM11690
      CALL IRANDP(1,NMEAS,IA,IA,IA,IA,IA,IA,4)                          EAM11700
      CALL RANDPD(1,PSCALE,DA,DA,DA,DA,DA,DA,4)                         EAM11710
      RETURN                                                            EAM11720
C                                                                       EAM11730
C     INTERROGATE FIGURE SENSOR AND SCALE MEASUREMENT                   EAM11740
C     SCALE POSITION                                                    EAM11750
  2   DA=0.0                                                            EAM11760
      DO 2000 I=1,NMEAS                                                 EAM11770
      X=XFSV(I)*PSCALE                                                  EAM11780
      Y=YFSV(I)*PSCALE                                                  EAM11790
C                                                                       EAM11800
C                                                                       EAM11810
C     INSERT DTOA AND ATOD SOFTWARE HERE                                EAM11820
C     (DA=DA+MEASURED FIGURE ERROR)                                     EAM11830
C                                                                       EAM11840
C                                                                       EAM11850
 2000 CONTINUE                                                          EAM11860
      XFV(I)=DA*FSCALV(I)/NMEAS                                         EAM11870
      RETURN                                                            EAM11880
C                                                                       EAM11890
      END                                                               EAM11900
```

```
      SUBROUTINE INIT(NENTRY)                                          EAM11910
C                                                                      EAM11920
C     SUBROUTINE TO INITIALIZE THE EXPERIMENTAL ACTIVE MIRROR          EAM11930
C                                                                      EAM11940
      DIMENSION XV(10000),AM(3000),AIM(3000),XFV(20),UFV(20),ASCALV(20),EAM11950
     1 FSCALV(20),XFSV(20),YFSV(20),XFRV(20),DUMV(20),UFAV(20),DUMVA(20)EAM11960
     2 ,GAINV(10),GAINM(400),LACTV(20)                                 EAM11970
      COMMON/BLKMFC/ N,NR,XV,MODE,LACTV,MODOP,NSNSWT,NRA               EAM11980
      EQUIVALENCE (XV(1),AM(1)),(XV(3001),AIM(1)),(XV(6001),XFV(1)),   EAM11990
     1 (UFV(1),XV(6101)),(ASCALV(1),XV(6121)),(FSCALV(1),XV(6141)),    EAM12000
     2 (XFSV(1),XV(6161)),(YFSV(1),XV(6181)),(XFRV(1),XV(6201)),       EAM12010
     3 (UFAV(1),XV(6221)),(DUMV(1),XV(6241)),(DUMVA(1),XV(6261)),      EAM12020
     4 (GAINV(1),XV(6281)),(GAINM(1)(XV(6291))                         EAM12030
C                                                                      EAM12040
 1000 FORMAT(4HINIT)                                                   EAM12050
C                                                                      EAM12060
      GO TO (1,2),NENTRY                                               EAM12070
C                                                                      EAM12080
C     READ DATA FOR TYPEWRITER CONTROL                                 EAM12090
 1    PRINT 1000                                                       EAM12100
C                                                                      EAM12110
C     READ BASIC DATA FOR THE EXPERIMENTAL ACTIVE MIRROR               EAM12120
C     N,NR                                                             EAM12130
      CALL IRANDP(2,N,NR,IA,IA,IA,IA,IA,4)                             EAM12140
C     AM                                                               EAM12150
      CALL MXRNP(AM,N,N,4)                                             EAM12160
C     FSCALV                                                           EAM12170
      CALL MXRNP(FSCALV,1,N,4)                                         EAM12180
C     XFSV                                                             EAM12190
      CALL MXRNP(XFSV,1,N,4)                                           EAM12200
C     YFSV                                                             EAM12210
      CALL MXRNP(YFSV,1,N,4)                                           EAM12220
C     LACTV                                                            EAM12230
      CALL IMXRNP(LACTV,1,N,4)                                         EAM12240
C     ASCALV                                                           EAM12250
      CALL MXRNP(ASCALV,1,NR,4)                                        EAM12260
      CALL FIGSEN(1,I,XFV,XFSV,YFSV,FSCALV)                            EAM12270
      CALL ACTCMD(1,I,UFV,UFAV,ASCALV)                                 EAM12280
      CALL MIRCAL(1)                                                   EAM12290
      CALL ACTCAL(1)                                                   EAM12300
C                                                                      EAM12310
      RETURN                                                           EAM12320
C                                                                      EAM12330
C     INITIALIZE MIRROR FIGURE CONTROL SYSTEM                          EAM12340
 2    DO 2201 I=1,N                                                    EAM12350
      XFV(I)=0.0                                                       EAM12360
      XFRV(I)=0.0                                                      EAM12370
      DUMV(I)=0.0                                                      EAM12380
      DUMVA(I)=0.0                                                     EAM12390
 2201 UFV(I)=0.0                                                       EAM12400
      DO 2202 I=1,NR                                                   EAM12410
 2202 CALL ACTCMD(2,I,UFV,UFAV,ASCALV)                                 EAM12420
      DO 2203 I=1,N                                                    EAM12430
 2203 CALL FIGSEN(2,I,XFV,XFSV,YFSV,FSCALV)                            EAM12440
      RETURN                                                           EAM12450
C                                                                      EAM12460
      END                                                              EAM12470
```

```
      SUBROUTINE MFCS(NENTRY)                                           EAM12480
C                                                                       EAM12490
C     SUBROUTINE TO REALIZE MIRROR FIGURE CONTROL SYSTEMS               EAM12500
C                                                                       EAM12510
      DIMENSION XV(10000),AM(3000),AIM(3000),XFV(20),UFV(20),ASCALV(20),EAM12520
     1 FSCALV(20),XFSV(20),YFSV(20),XFRV(20),DUMV(20),UFAV(20),DUMVA(20)EAM12530
     2 ,GAINV(10),GAINM(400),LACTV(20)                                  EAM12540
      COMMON/BLKMFC/ N,NR,XV,MODE,LACTV,MODOP,NSNSWT,NRA                EAM12550
      EQUIVALENCE (XV(1),AM(1)),(XV(3001),AIM(1)),(XV(6001),XFV(1)),    EAM12560
     1 (UFV(1),XV(6101)),(ASCALV(1),XV(6121)),(FSCALV(1),XV(6141)),     EAM12570
     2 (XFSV(1),XV(6161)),(YFSV(1),XV(6181)),(XFRV(1),XV(6201)),        EAM12580
     3 (UFAV(1),XV(6221)),(DUMV(1),XV(6241)),(DUMVA(1),XV(6261)),       EAM12590
     4 (GAINV(1),XV(6281)),(GAINM(1)(XV(6291))                          EAM12600
      DIMENSION PARV(20),GRADV(20)                                      EAM12610
      COMMON/BLKA/PINDEX,PISTOR,PARV,EPS,GRADV,NIC,NHC,NPAR,NIM,NHM     EAM12620
                                                                        EAM12630
C     DIMENSION SPARV(20),DV(20)                                        EAM12640
      COMMON/BLKA1/NOPT,EPSINC,EPSDEC,ISDEC,NSDEC,SNHC,SPARV,DV         EAM12650
C                                                                       EAM12660
C                                                                       EAM12670
C                                                                       EAM12680
 1000 FORMAT(4HMFCS)                                                    EAM12690
 1010 FORMAT(1H1,/,10X,5HGAINM)                                         EAM12700
C                                                                       EAM12710
      GO TO (1,2,3,4),NENTRY                                            EAM12720
C                                                                       EAM12730
C     MIRROR FIGURE CONTROL SYSTEM INITIALIZATION                       EAM12740
    1 PRINT 1000                                                        EAM12750
      CALL MXRNP(GAINV,1,2,4)                                           EAM12760
      CALL RANDPD(1,DTITCS,DA,DA,DA,DA,DA,DA,4)                         EAM12770
      IA=MODOP-5                                                        EAM12780
C                                                                       EAM12790
C     CALCULATE FEEDBACK MATRIX                                         EAM12800
      GO TO(2110,2120,2130),IA                                          EAM12810
C                                                                       EAM12820
C     SLCS                                                              EAM12830
C     GAINM=ARR**-1                                                     EAM12840
C     GENERATE ARR                                                      EAM12850
 2110 NRA=NR                                                            EAM12860
      CALL REDUAM(1)                                                    EAM12870
      CALL REDUAM(2)                                                    EAM12880
C     GAINM=ARR**-1                                                     EAM12890
      CALL SINV(NR,AM,GAINM,DA)                                         EAM12900
C                                                                       EAM12910
      PRINT SLCS GAIN MATRIX                                            EAM12920
      PRINT 1010                                                        EAM12930
      CALL MXRNP(GAINM,NR,NR,3)                                         EAM12940
      RETURN                                                            EAM12950
C                                                                       EAM12960
C     LOCS                                                              EAM12970
C     GAINM=ART*AR                                                      EAM12980
 2120 NRA=N                                                             EAM12990
      CALL MTRA(AM,AIM,N,NR,0)                                          EAM13000
      CALL MPRD(AIM,AM,GAINM,NR,N,0,0,NR)                               EAM13010
C     GAINM=((ART*AR)**-1)*ART                                          EAM13020
      CALL SINV(NR,GAINM,AM,DA)                                         EAM13030
      CALL MPRD(AM,AIM,GAINM,NR,NR,0,0,N)                               EAM13040
```

```
C                                                                       EAM13050
C     PRINT LOCS GAIN MATRIX                                            EAM13060
      PRINT 1010                                                        EAM13070
      CALL MXRNP(GAINM,NR,N,3)                                          EAM13080
      RETURN                                                            EAM13090
C                                                                       EAM13100
C     INITIALIZE ITERATIVE CONTROL SYSTEM                               EAM13110
 2130 CONTINUE                                                          EAM13120
      RETURN                                                            EAM13130
C                                                                       EAM13140
C     CALCULATE FIGURE CONTROLS                                         EAM13150
C     ACTUATOR INPUT COMPUTATION                                        EAM13160
C     MEASURE FIGURE ERROR                                              EAM13170
 2    DO 2505 I=1,N                                                     EAM13180
 2505 CALL FIGSEN(2,I,XFV,XFSV,YFSV,FSCALV)                             EAM13190
C                                                                       EAM13200
C     GENERATE XFRV                                                     EAM13210
      IA=MODOP-5                                                        EAM13220
      J=0                                                               EAM13230
      DO 2506 I=1,N                                                     EAM13240
      GO TO (2508,2509),IA                                              EAM13250
 2508 IF(LACTV(I))2507,2506,2507                                        EAM13260
 2507 J=J+1                                                             EAM13270
      XFRV(J)=XFV(I)                                                    EAM13280
      GO TO 2506                                                        EAM13290
 2509 XFRV(I)=XFV(I)                                                    EAM13300
 2506 CONTINUE                                                          EAM13310
C                                                                       EAM13320
C     DUMV=GAINM*XFRV                                                   EAM13330
      CALL MPRD(GAINM,XFRV,DUMV,NR,NRA,0,0,1)                           EAM13340
C     DUMV=DUMV*GAINV(1)                                                EAM13350
      DO 2510 I=1,NR                                                    EAM13360
 2510 DUMV(I)=DUMV(I)*GAINV(1)                                          EAM13370
C     INTEGRAL COMPENSATION                                             EAM13380
C     DUMVA=DUMV*DT+DUMVA                                               EAM13390
      DO 2520 I=1,NR                                                    EAM13400
 2520 DUMVA(I)=DUMV(I)*DT+DUMVA(I)                                      EAM13410
C                                                                       EAM13420
C     ACTUATOR OUTPUT CONTROL SYSTEM                                    EAM13430
C     UFV=GAINV(2)*(DUMVA-UFAV)*DT-UFV                                  EAM13440
 3    DO 2530 I=1,NR                                                    EAM13450
 2530 UFV(I)=GAINV(2)*(DUMVA(I)-UFAV(I))*DT+UFV(I)                      EAM13460
C                                                                       EAM13470
C     TRANSFER COMMANDS TO ACTUATORS AND RETURN ACTUATOR OUTPUTS        EAM13480
      DO 2540 I=1,NR                                                    EAM13490
 2540 CALL ACTCMD(2,I,UFV,UFAV,ASCALV)                                  EAM13500
      RETURN                                                            EAM13510
C                                                                       EAM13520
C     ITCS ACTUATOR COMMANDS                                            EAM13530
C                                                                       EAM13540
C     ITCS COMMANDS                                                     EAM13550
 4    CALL REALT(TREAL)                                                 EAM13560
      DA=TREAL                                                          EAM13570
      DB=TREAL+DTITCS                                                   EAM13580
C     DUMVA=PARV                                                        EAM13590
      DO 2550 I=1,NR                                                    EAM13600
 2550 DUMVA(I)=PARV(I)                                                  EAM13610
```

```
C                                                                       EAM13620
C       ACTUATOR OUTPUT CONTROL SYSTEM                                  EAM13630
C       UFV=GAIN(2)*(DUMVA-UFAV)*DT-UFV                                 EAM13640
 2554   DO 2551 I=1,NR                                                  EAM13650
        UFV(I)=GAINV(2)*(DUMVA(I)-UFAV(I))*DT+UFV(I)                    EAM13660
 2551   CALL ACTCMD(2,I,UFV,UFAV,ASCALV)                                EAM13670
        DA=DA+DT                                                        EAM13680
 2552   CALL REALT(TREAL)                                               EAM13690
        IF(TREAL-DB)2555,2553,2553                                      EAM13700
 2555   IF(TREAL-DA)2552,2554,2554                                      EAM13710
 2553   RETURN                                                          EAM13720
C                                                                       EAM13730
        END                                                             EAM13740
```

```
      SUBROUTINE MIRCAL(NENTRY)                                          EAM13750
C                                                                        EAM13760
C     SUBROUTINE TO CALIBRATE MIRROR                                     EAM13770
C                                                                        EAM13780
      DIMENSION XV(10000),AM(3000),AIM(3000),XFV(20),UFV(20),ASCALV(20),EAM13790
     1 FSCALV(20),XFSV(20),YFSV(20),XFRV(20),DUMV(20),UFAV(20),DUMVA(20)EAM13800
     2 ,GAINV(10),GAINM(400),LACTV(20)                                  EAM13810
      COMMON/BLKMFC/ N,NR,XV,MODE,LACTV,MODOP,NSNSWT,NRA                 EAM13820
      EQUIVALENCE (XV(1),AM(1)),(XV(3001),AIM(1)),(XV(6001),XFV(1)),     EAM13830
     1 (UFV(1),XV(6101)),(ASCALV(1),XV(6121)),(FSCALV(1),XV(6141)),      EAM13840
     2 (XFSV(1),XV(6161)),(YFSV(1),XV(6181)),(XFRV(1),XV(6201)),         EAM13850
     3 (UFAV(1),XV(6221)),(DUMM(1),XV(6241)),(DUMVA(1),XV(6261)),        EAM13860
     4 (GAINV(1),XV(6281)),(GAINM(1)(XV(6291))                          EAM13870
C                                                                        EAM13880
C                                                                        EAM13890
      DIMENSION DUMM(20)                                                 EAM13900
C                                                                        EAM13910
 1000 FORMAT(6HMIRCAL)                                                   EAM13920
 1001 FORMAT(1H1/13X,2HAM)                                               EAM13930
 1002 FORMAT(8HMCAL END)                                                 EAM13940
C                                                                        EAM13950
      GO TO (1,2),NENTRY                                                 EAM13960
C                                                                        EAM13970
C     INPUT DATA                                                         EAM13980
 1    PRINT 1000                                                         EAM13990
      CALL IRANDP(1,NMEAS,IA,IA,IA,IA,IA,IA,4)                           EAM14000
      CALL RANDPD(1,DACT,DA,DA,DA,DA,DA,DA,4)                            EAM14010
      DB=2.0*DACT                                                        EAM14020
      RETURN                                                             EAM14030
C                                                                        EAM14040
C     MIRROR CALIBRATION                                                 EAM14050
 2    CALL INIT(2)                                                       EAM14060
      DO 2200 I=1,NR                                                     EAM14070
      DO 2201 J=1,NMEAS                                                  EAM14080
      UFV(I)=-DACT                                                       EAM14090
      CALL ACTCMD(2,I,UFV,UFAV,ASCALV)                                   EAM14100
      DO 2202 K=1,N                                                      EAM14110
      CALL FIGSEN(2,K,XFV,XFSV,YFSV,FSCALV)                              EAM14120
 2202 DUMV(K)=XFV(K)                                                     EAM14130
      UFV(I)=DACT                                                        EAM14140
      CALL ACTCMD(2,I,UFV,UFAV,ASCALV)                                   EAM14150
      DO 2203 K=1,N                                                      EAM14160
      CALL FIGSEN(2,K,XFV,XFSV,YFSV,FSCALV)                              EAM14170
      DUMV(K)=(XFV(K)-DUMV(K))/DB                                        EAM14180
 2203 AM(K+(I-1)*N)=DUMV(K)+AM(K+(I-1)*N)                                EAM14190
 2201 CONTINUE                                                           EAM14200
      DO 2204 K=1,N                                                      EAM14210
 2204 AM(K+(I-1)*N)=AM(K+(I-1)*N)/NMEAS                                  EAM14220
 2200 CONTINUE                                                           EAM14230
C                                                                        EAM14240
C     PRINT OUT MIRROR DEFORMATION-ACTUATOR COMMAND ARRAY                EAM14250
      PRINT 1001                                                         EAM14260
      CALL MXRNP(AM,N,NR,3)                                              EAM14270
      TYPE 1002                                                          EAM14280
      RETURN                                                             EAM14290
C                                                                        EAM14300
      END                                                                EAM14310
```

```
      SUBROUTINE MODCHK(NENTRY,MODE,MODED,MODOP,MODOPD,ITEST)          EAM14320
C                                                                      EAM14330
C     SUBROUTINE TO CHECK VALUES OF MODE AND MODOP                     EAM14340
C                                                                      EAM14350
 1000 FORMAT(4HMODE,I6,5H NOT=,I5)                                     EAM14360
 1010 FORMAT(5HMODOP,I5,5H NOT=,I5)                                    EAM14370
C                                                                      EAM14380
C     CHECK MODE                                                       EAM14390
      IF(MODE-MODED)2020,2010,2020                                     EAM14400
 2020 TYPE 1000,MODE,MODED                                             EAM14410
      ITEST=2                                                          EAM14420
C                                                                      EAM14430
C     CHECK MODOP                                                      EAM14440
 2010 IF(MODOP-MODOPD)2030,2040,2030                                   EAM14450
 2030 TYPE 1010,MODOP,MODOPD                                           EAM14460
      ITEST=2                                                          EAM14470
 2040 RETURN                                                           EAM14480
C                                                                      EAM14490
      END                                                              EAM14500
```

```
      SUBROUTINE REALT(TREAL)                                          EAM14510
C                                                                      EAM14520
C     SUBROUTINE TO INTERROGATE REAL TIME CLOCK                        EAM14530
C                                                                      EAM14540
C     TREAL=REAL TIME                                                  EAM14550
C                                                                      EAM14560
C                                                                      EAM14570
C     INSERT REAL TIME CLOCK INTERROGATION SOFTWARE HERE               EAM14580
C                                                                      EAM14590
C                                                                      EAM14600
      RETURN                                                           EAM14610
C                                                                      EAM14620
      END                                                              EAM14630
```

```
      SUBROUTINE REDUAM(NENTRY)                                          EAM14640
C                                                                        EAM14650
C     SUBROUTINE TO GENERATE AR AND ARR FROM A                           EAM14660
C                                                                        EAM14670
      DIMENSION XV(10000),AM(3000),AIM(3000),XFV(20),UFV(20),ASCALV(20),EAM14680
     1 FSCALV(20),XFSV(20),YFSV(20),XFRV(20),DUMV(20),UFAV(20),DUMVA(20)EAM14690
     2 ,GAINV(10),GAINM(400),LACTV(20)                                  EAM14700
      COMMON/BLKMFC/ N,NR,XV,MODE,LACTV,MODOP,NSNSWT,NRA                 EAM14710
      EQUIVALENCE (XV(1),AM(1)),(XV(3001),AIM(1)),(XV(6001),XFV(1)),     EAM14720
     1 (UFV(1),XV(6101)),(ASCALV(1),XV(6121)),(FSCALV(1),XV(6141)),      EAM14730
     2 (XFSV(1),XV(6161)),(YFSV(1),XV(6181)),(XFRV(1),XV(6201)),         EAM14740
     3 (UFAV(1),XV(6221)),(DUMV(1),XV(6241)),(DUMVA(1),XV(6261)),        EAM14750
     4 (GAINV(1),XV(6281)),(GAINM(1)(XV(6291))                          EAM14760
C                                                                        EAM14770
 1000 FORMAT(1H1,/,13X,2HAR)                                             EAM14780
 1010 FORMAT(1H1,/,12X,3HARR)                                            EAM14790
C                                                                        EAM14800
      GO TO(1,2),NENTRY                                                  EAM14810
C                                                                        EAM14820
C     GENERATE AR BY REMOVING COLUMNS FROM A                             EAM14830
  1   K=0                                                                EAM14840
      DO 2000 J=1,N                                                      EAM14850
      IF(LACTV(J))2010,2000,2010                                         EAM14860
 2010 K=K+1                                                              EAM14870
      DO 2020 I=1,N                                                      EAM14880
      CALL LOC(I,J,IA,N,N,0)                                             EAM14890
      CALL LOC(I,K,IB,N,N,0)                                             EAM14900
 2020 AIM(IB)=AM(IA)                                                     EAM14910
 2000 CONTINUE                                                           EAM14920
C                                                                        EAM14930
C     COPY RESULT INTO AM                                                EAM14940
      CALL MCPY(AIM,AM,N,NR,0)                                           EAM14950
C                                                                        EAM14960
C     PRINT AR                                                           EAM14970
      PRINT 1000                                                         EAM14980
      CALL MXRNP(AM,N,NR,3)                                              EAM14990
      RETURN                                                             EAM15000
C                                                                        EAM15010
C     GENERATE ARR FROM AR BY REMOVING ROWS FROM AR                      EAM15020
  2   K=0                                                                EAM15030
      DO 2100 I=1,N                                                      EAM15040
      IF(LACTV(I))2110,2100,2110                                         EAM15050
 2110 K=K+1                                                              EAM15060
      DO 2120 J=1,NR                                                     EAM15070
      CALL LOC(I,J,IA,N,NR,0)                                            EAM15080
      CALL LOC(K,J,IB,NR,NR,0)                                           EAM15090
 2120 AIM(IB)=AM(IA)                                                     EAM15100
 2100 CONTINUE                                                           EAM15110
C                                                                        EAM15120
C     COPY RESULT INTO AM                                                EAM15130
      CALL MCPY(AIM,AM,NR,NR,0)                                          EAM15140
C                                                                        EAM15150
C     PRINT ARR                                                          EAM15160
      PRINT 1010                                                         EAM15170
      CALL MXRNP(AM,NR,NR,3)                                             EAM15180
      RETURN                                                             EAM15190
C                                                                        EAM15200
      END                                                                EAM15210
```

```
      SUBROUTINE TYPCON(NENTRY)                                          EAM15220
C                                                                        EAM15230
C     SUBROUTINE FOR TYPEWRITER CONTROL OF THE EXPERIMENTAL ACTIVE MIRROEAM15240
C                                                                        EAM15250
      DIMENSION XV(10000),AM(3000),AIM(3000),XFV(20),UFV(20),ASCALV(20),EAM15260
     1 FSCALV(20),XFSV(20),YFSV(20),XFRV(20),DUMV(20),UFAV(20),DUMVA(20)EAM15270
     2 ,GAINV(10),GAINM(400),LACTV(20)                                   EAM15280
      COMMON/BLKMFC/ N,NR,XV,MODE,LACTV,MODOP,NSNSWT,NRA                 EAM15290
      EQUIVALENCE (XV(1),AM(1)),(XV(3001),AIM(1)),(XV(6001),XFV(1)),     EAM15300
     1 (UFV(1),XV(6101)),(ASCALV(1),XV(6121)),(FSCALV(1),XV(6141)),      EAM15310
     2 (XFSV(1),XV(6161)),(YFSV(1),XV(6181)),(XFRV(1),XV(6201)),         EAM15320
     3 (UFAV(1),XV(6221)),(DUMV(1),XV(6241)),(DUMVA(1),XV(6261)),        EAM15330
     4 (GAINV(1),XV(6281)),(GAINM(1)(XV(6291))                           EAM15340
C                                                                        EAM15350
      DIMENSION PARV(20),GRADV(20)                                       EAM15360
      COMMON/BLKA/PINDEX,PISTOR,PARV,EPS,GRADV,NIC,NHC,NPAR,NIM,NHM      EAM15370
C                                                                        EAM15380
      DIMENSION NAMV(20)                                                 EAM15390
C                                                                        EAM15400
       DATA NAMV /4HXV   ,4HAM   ,4HAIM ,4HXFV ,4HUFV ,4HUFAV,4HASCV,    EAM15410
      1 4HFSCV,4HXFSV,4HYFSV,4HXFRV,4HDUMV,4HLACT,4HGANM,4HGANV,4HLACV,  EAM15420
     2 4HDEND/                                                           EAM15430
C                                                                        EAM15440
 1000 FORMAT(A4)                                                         EAM15450
 1001 FORMAT(/6HNAMERR)                                                  EAM15460
 1002 FORMAT(10I3)                                                       EAM15470
 1003 FORMAT(3HNAM)                                                      EAM15480
 1004 FORMAT(4HINDX)                                                     EAM15490
 1005 FORMAT(10F12.6)                                                    EAM15500
 1006 FORMAT(4HMODE)                                                     EAM15510
 1007 FORMAT(4HINIT)                                                     EAM15520
 1008 FORMAT(4HSTRT)                                                     EAM15530
 1009 FORMAT(6HTYPCON)                                                   EAM15540
 1010 FORMAT(6HACTTST)                                                   EAM15550
 1011 FORMAT(6HMIRTST)                                                   EAM15560
 1012 FORMAT(6HMODFIN)                                                   EAM15570
 1013 FORMAT(4HSLCS)                                                     EAM15580
 1014 FORMAT(4HLOCS)                                                     EAM15590
 1015 FORMAT(4HITCS)                                                     EAM15600
 1016 FORMAT(1HJ)                                                        EAM15610
 1017 FORMAT(2HJM)                                                       EAM15620
 1018 FORMAT(4HCHNG)                                                     EAM15630
 1019 FORMAT(9HNEW VALUE)                                                EAM15640
 1020 FORMAT(7HTOO BIG)                                                  EAM15650
C                                                                        EAM15660
C                                                                        EAM15670
      GO TO (1,2),NENTRY                                                 EAM15680
C                                                                        EAM15690
C     INITIALIZATION                                                     EAM15700
  1   PRINT 1009                                                         EAM15710
      NNAMV=14                                                           EAM15720
      ICHNG=2                                                            EAM15730
      RETURN                                                             EAM15740
C                                                                        EAM15750
C     OPERATION                                                          EAM15760
  2   CONTINUE                                                           EAM15770
C                                                                        EAM15780
```

```
C      SELECT EXPERIMENTAL MODE                                          EAM15790
 2280  TYPE 1006                                                         EAM15800
       ACCEPT 1002,MODE                                                  EAM15810
       TYPE 1002,MODE                                                    EAM15820
       GO TO (2211,2212,2213,2214,2215,2216,2217,2218,2219,2220,2221),   EAM15830
      1 MODE                                                             EAM15840
C                                                                        EAM15850
C      INITIALIZE MFCS                                                   EAM15860
 2211  TYPE 1007                                                         EAM15870
       GO TO 2207                                                        EAM15880
C                                                                        EAM15890
C      START MFCS                                                        EAM15900
 2212  TYPE 1008                                                         EAM15910
       GO TO 2207                                                        EAM15920
C                                                                        EAM15930
C      DIAGNOSTIC MODE                                                   EAM15940
 2213  GO TO 2990                                                        EAM15950
C                                                                        EAM15960
C      TEST ACTUATORS                                                    EAM15970
 2214  TYPE 1010                                                         EAM15980
       CALL ACTCAL(2)                                                    EAM15990
       GO TO 2280                                                        EAM16000
C                                                                        EAM16010
C      TEST MIRROR                                                       EAM16020
 2215  TYPE 1010                                                         EAM16030
       CALL MIRCAL(2)                                                    EAM16040
       GO TO 2280                                                        EAM16050
C                                                                        EAM16060
C      SIMPLIFIED LINEAR FIGURE CONTROL                                  EAM16070
 2216  TYPE 1013                                                         EAM16080
       MODOP=6                                                           EAM16090
       GO TO 2207                                                        EAM16100
C                                                                        EAM16110
C      LINEAR OPTIMAL FIGURE CONTROL                                     EAM16120
 2217  TYPE 1014                                                         EAM16130
       MODOP=7                                                           EAM16140
       GO TO 2207                                                        EAM16150
C                                                                        EAM16160
C      ITERATIVE FIGURE CONTROL                                          EAM16170
 2218  TYPE 1015                                                         EAM16180
       MODOP=8                                                           EAM16190
       GO TO 2207                                                        EAM16200
C                                                                        EAM16210
C      EVALUATE AND TYPE FIGURE PERFORMANCE INDEX                        EAM16220
 2219  CALL PINDX(2)                                                     EAM16230
       TYPE 1016                                                         EAM16240
       TYPE 1005,PINDEX                                                  EAM16250
       DA=SQRT(PINDEX/N)                                                 EAM16260
       TYPE 1017                                                         EAM16270
       TYPE 1005,DA                                                      EAM16280
       GO TO 2280                                                        EAM16290
C                                                                        EAM16300
C      MODIFY DATA BUSS VALUE                                            EAM16310
 2220  ICHNG=2                                                           EAM16320
       TYPE 1018                                                         EAM16330
       GO TO 2990                                                        EAM16340
C                                                                        EAM16350
```

```
C     REQUEST MODE AGAIN IF MODE VALUE IS TOO LARGE                    EAM16360
 2221 TYPE 1020                                                        EAM16370
      GO TO 2280                                                       EAM16380
C                                                                      EAM16390
C     IDENTIFY VARIABLE NAME                                           EAM16400
 2990 TYPE 1003                                                        EAM16410
      ACCEPT 1000,KK                                                   EAM16420
      TYPE 1000,KK                                                     EAM16430
      DO 2302 I=1,NNAMV                                                EAM16440
      IF(KK-NAMV(I))2302,2301,2302                                     EAM16450
 2301 LL=I                                                             EAM16460
      GO TO 2303                                                       EAM16470
 2302 CONTINUE                                                         EAM16480
      TYPE 1001                                                        EAM16490
      GO TO 2990                                                       EAM16500
 2303 TYPE 1000,KK                                                     EAM16510
C                                                                      EAM16520
C     IDENTIFY VARIABLE INDEX                                          EAM16530
      TYPE 1004                                                        EAM16540
      GO TO (2310,2320,2320,2310,2310,2310,2310,2310,2310,2310,2310,   EAM16550
     1 2310,2320,2310,2310,2280),LL                                    EAM16560
 2310 ACCEPT 1002,II                                                   EAM16570
      TYPE 1002,II                                                     EAM16580
      GO TO 2330                                                       EAM16590
 2320 ACCEPT 1002,II,JJ                                                EAM16600
      TYPE 1002,II,JJ                                                  EAM16610
      GO TO 2330                                                       EAM16620
C                                                                      EAM16630
C     ACCEPT NEW VALUE IF ICHNG=2                                      EAM16640
 2330 GO TO(2331,2340),ICHNG                                           EAM16650
 2331 ACCEPT 1005,V                                                    EAM16660
      TYPE 1019                                                        EAM16670
      TYPE 1005,V                                                      EAM16680
      GO TO 2340                                                       EAM16690
C                                                                      EAM16700
C     TYPE VALUE OF INDEXED VARIABLE                                   EAM16710
 2340 GO TO (2401,2402,2403,2404,2405,2406,2407,2408,2409,2410,2411,   EAM16720
     1 2412,2413,2414,2415,2280),LL                                    EAM16730
 2401 CALL CHNG(ICHNG,V,XV(II))                                        EAM16740
      GO TO 2500                                                       EAM16750
 2402 CALL ELMA(ICHNG,AM,II,JJ,V,NR)                                   EAM16760
      CALL ELMA(2,AM,II,JJ,V,NR)                                       EAM16770
      TYPE 1005,V                                                      EAM16780
      GO TO 2500                                                       EAM16790
 2403 CALL ELMA(ICHNG,AIM,II,JJ,V,NR)                                  EAM16800
      CALL ELMA(2,AIM,II,JJ,V,NR)                                      EAM16810
      TYPE 1005,V                                                      EAM16820
      GO TO 2500                                                       EAM16830
 2404 CALL CHNG(ICHNG,V,XFV(II))                                       EAM16840
      TYPE 1005,XFV(II)                                                EAM16850
      GO TO 2500                                                       EAM16860
 2405 CALL CHNG(ICHNG,V,XFRV(II))                                      EAM16870
      TYPE 1005,XFRV(II)                                               EAM16880
      GO TO 2500                                                       EAM16890
 2406 CALL CHNG(ICHNG,V,UFV(II))                                       EAM16900
      TYPE 1005,UFV(II)                                                EAM16910
      GO TO 2500                                                       EAM16920
```

```
 2407 CALL CHNG(ICHNG,V,UFAV(II))                                       EAM16930
      TYPE 1005,UFAV(II)                                                EAM16940
      GO TO 2500                                                        EAM16950
 2408 CALL CHNG(ICHNG,V,ASCALV(II))                                     EAM16960
      TYPE 1005,ASCALV(II)                                              EAM16970
      GO TO 2500                                                        EAM16980
 2409 CALL CHNG(ICHNG,V,FSCALV(II))                                     EAM16990
      TYPE 1005,FSCALV(II)                                              EAM17000
      GO TO 2500                                                        EAM17010
 2410 CALL CHNG(ICHNG,V,XFSV(II))                                       EAM17020
      TYPE 1005,XFSV(II)                                                EAM17030
      GO TO 2500                                                        EAM17040
 2411 CALL CHNG(ICHNG,V,YFSV(II))                                       EAM17050
      TYPE 1005,YFSV(II)                                                EAM17060
      GO TO 2500                                                        EAM17070
 2412 CALL CHNG(ICHNG,V,DUMV(II))                                       EAM17080
      TYPE 1005, DUMV(II)                                               EAM17090
      GO TO 2500                                                        EAM17100
 2413 CALL ELMA(2,GAINM,II,JJ,V,NR)                                     EAM17110
      TYPE 1005,V                                                       EAM17120
 2414 TYPE 1005,GAINV(II)                                               EAM17130
      GO TO 2500                                                        EAM17140
 2415 TYPE 1002,LACTV(II)                                               EAM17150
      GO TO 2500                                                        EAM17160
 2500 ICHNG=1                                                           EAM17170
      GO TO 2280                                                        EAM17171
 2207 RETURN                                                            EAM17180
C                                                                       EAM17190
      END                                                               EAM17200
```

PRECEDING PAGE BLANK NOT FILMED

APPENDIX B
PARAMETER OPTIMIZATION PACKAGE

B.0 Introduction

This appendix contains listings for the following subroutines.

```
      SUBROUTINE EPCHNG(NENTRY,STEP,STPDEC,DJDPV)                       POP10000
C                                                                       POP10010
C     SUBROUTINE TO LIMIT THE ABSOLUTE VALUE OF THE CHANGE IN THE       POP10020
C     COMPONENTS OF THE PARAMETER VECTOR                                POP10030
C                                                                       POP10040
      DIMENSION DJDPV(1)                                                POP10050
C                                                                       POP10060
      DIMENSION PARV(20),GRADV(20)                                      POP10070
      COMMON/BLKA/PINDEX,PISTOR,PARV,EPS,GRADV,NIC,NHC,NPAR,NIM,NHM     POP10080
C                                                                       POP10090
      DIMENSION DPARLV(20),PARMIN(20)                                   POP10100
C                                                                       POP10110
 1000 FORMAT(6HEPCHNG)                                                  POP10120
C                                                                       POP10130
      GO TO(1,2),NENTRY                                                 POP10140
C                                                                       POP10150
C     INPUT DATA                                                        POP10160
    1 PRINT 1000                                                        POP10170
C     READ IN MAXIMUM ABSOLUTE CHANGE IN THE EACH ELEMENT OF THE        POP10180
C     PARAMETER VECTOR                                                  POP10190
      CALL MXRNP(DPARLV,1,NPAR,4)                                       POP10200
C     READ IN MINIMUM PARAMETER VALUE WITH SIGN                         POP10210
      CALL MXRNP(PARMIN,1,NPAR,4)                                       POP10220
      RETURN                                                            POP10230
C                                                                       POP10240
C     SELECT EPS BASED ON ABSOLUTE VALUE OF CHANGE IN PARAMETER         POP10250
    2 DO 2000 I=1,NPAR                                                  POP10260
 2005 DA=DJDPV(I)*STEP                                                  POP10270
      IF(ABS(DA)-DPARLV(I))2020,2020,2010                               POP10280
 2010 NHC=NHC+1                                                         POP10290
      STEP=STEP*STPDEC                                                  POP10300
      GO TO 2005                                                        POP10310
C                                                                       POP10320
 2020 DB=PARV(I)-DA                                                     POP10330
      IF(PARMIN(I))2021,2000,2021                                       POP10340
 2021 IF((DB-PARMIN(I))*PARMIN(I)/ABS(PARMIN(I)))2030,2000,2000         POP10350
 2030 DJDPV(I)=0.0                                                      POP10360
      PARV(I)=PARMIN(I)                                                 POP10370
 2000 CONTINUE                                                          POP10380
      CALL MFCS(4)                                                      POP10390
      RETURN                                                            POP10400
C                                                                       POP10410
      END                                                               POP10420
```

```
      SUBROUTINE GRAD(NENTRY)                                          POP10430
C                                                                      POP10440
C     SUBROUTINE TO CALCULATE GRADIENT VECTOR                          POP10450
C     SPECIAL VERSION FOR OPTIMIZING THE PARAMETERS IN A DYNAMICAL SYSTEPOP10460
C                                                                      POP10470
      DIMENSION PARV(20),GRADV(20)                                     POP10480
      COMMON/BLKA/PINDEX,PISTOR,PARV,EPS,GRADV,NIC,NHC,NPAR,NIM,NHM    POP10490
C                                                                      POP10500
 1001 FORMAT(4HGRAD)                                                   POP10510
C                                                                      POP10520
      GO TO (1,2),NENTRY                                               POP10530
C                                                                      POP10540
 1    PRINT 1001                                                       POP10550
      CALL RANDPD(1,DPAR,DA,DA,DA,DA,DA,DA,4)                          POP10560
      CALL IRANDP(1,NGRAD,JJ,JJ,JJ,JJ,JJ,JJ,4)                        POP10570
      CALL ANGRAD(1)                                                   POP10580
      RETURN                                                           POP10590
C                                                                      POP10600
C     IF SENSE SWITCH 5 IS SET CALCULATE THE GRADIENT NUMERICALLY      POP10610
 2    IF(SENSE SWITCH 5)3,7                                            POP10620
 7    GO TO(3,4,5),NGRAD                                               POP10630
C                                                                      POP10640
C     GRADIENT EVALUATION BY FINITE DIFFERENCES                        POP10650
 3    CALL PINDX(2)                                                    POP10660
      DA=PINDEX                                                        POP10670
      DO 1000 I=1,NPAR                                                 POP10680
      DB=PARV(I)                                                       POP10690
      PARV(I)=DB+DPAR                                                  POP10700
      CALL MFCS(4)                                                     POP10710
      CALL PINDX(2)                                                    POP10720
      GRADV(I)=(PINDEX-DA)/DPAR                                        POP10730
      PARV(I)=DB                                                       POP10740
 1000 CALL MFCS(4)                                                     POP10750
      RETURN                                                           POP10760
C                                                                      POP10770
C     GRADIENT COMPUTATION BY TWO POINT INTERPOLATION                  POP10780
 4    DC=2.0*DPAR                                                      POP10790
      DO 1100 I=1,NPAR                                                 POP10800
      DB=PARV(I)                                                       POP10810
      PARV(I)=DB+DPAR                                                  POP10820
      CALL MFCS(4)                                                     POP10830
      CALL PINDX(2)                                                    POP10840
      DA=PINDEX                                                        POP10850
      PARV(I)=DB-DPAR                                                  POP10860
      CALL MFCS(4)                                                     POP10870
      CALL PINDX(2)                                                    POP10880
      GRADV(I)=(DA-PINDEX)/DC                                          POP10890
      PARV(I)=DB                                                       POP10900
 1100 CALL MFCS(4)                                                     POP10910
      RETURN                                                           POP10920
C                                                                      POP10930
C     CALCULATE GRADIENT ANALYTICALLY                                  POP10940
 5    CALL ANGRAD(2)                                                   POP10950
      RETURN                                                           POP10960
C                                                                      POP10970
      END                                                              POP10980
```

```
      SUBROUTINE ITPRT(NENTRY)                                          POP10990
C                                                                       POP11000
C     SUBROUTINE TO PRINT OPTIMIZATION STATUS                           POP11010
C                                                                       POP11020
      DIMENSION PARV(20),GRADV(20)                                      POP11030
      COMMON/BLKA/PINDEX,PISTOR,PARV,EPS,GRADV,NIC,NHC,NPAR,NIM,NHM     POP11040
C                                                                       POP11050
      DIMENSION SPIV(20),SLGV(20)                                       POP11060
C                                                                       POP11070
 1000 FORMAT(/,12X,3HNIC,12X,3HNHC,9X,6HPISTOR,9X,6HPINDEX,9X,6HDPI/PI, POP11080
     1 12X,3HEPS,4X,11H/GRADV/PARV)                                     POP11090
 1010 FORMAT(2I15,4F15.6)                                               POP11100
 1040 FORMAT(5X,10H/SPIV/SLGV,6X,4HKKS=,I5)                             POP11110
 1050 FORMAT(3HSS3)                                                     POP11120
 1060 FORMAT(5HITPRT)                                                   POP11130
C                                                                       POP11140
      GO TO (1,2,3,4,5),NENTRY                                          POP11150
C                                                                       POP11160
C     ITERATION OUTPUT                                                  POP11170
 1    IA=NIC+NHC                                                        POP11180
      IF((NIC-NIM)*(NHC-NHM))2008,2020,2008                             POP11190
 2008 IF(SENSE SWITCH 2)2020,2009                                       POP11200
 2009 IF(IA-IKS)2010,2020,2020                                          POP11210
 2020 IKS=IA+NITPRT                                                     POP11220
C     PRINT NIC,NHC,PISTOR,PINDEX,DPI/PI,EPS,GRADV,PARV                 POP11230
      PRINT 1000                                                        POP11240
C     COMPUTE FRACTIONAL CHANGE IN PREVIOUS VALUE OF PERFORMANCE INDEX  POP11250
      DA=(PINDEX-PISTOR)/PISTOR                                         POP11260
      PRINT 1010,NIC,NHC,PISTOR,PINDEX,DA,EPS                           POP11270
      CALL MXRNP(GRADV,1,NPAR,3)                                        POP11280
      CALL MXRNP(PARV,1,NPAR,3)                                         POP11290
C     CALL TO PINDX FOR AUXILLIARY OUTPUT PRINT                         POP11300
      CALL PINDX(3)                                                     POP11310
C     PRINT OUT MODIFIED PERFORMANCE INDEX WEIGHTING FACTORS            POP11320
      CALL DESX(3)                                                      POP11330
C                                                                       POP11340
 2010 KK=NIC+1                                                          POP11350
      IF(KK-KKS)2000,2000,2001                                          POP11360
 2001 IA=KKS-20                                                         POP11370
C                                                                       POP11380
C     STORED INFORMATION OUTPUTED EVERY 20 ITERATIONS                   POP11390
      PRINT 1040,IA                                                     POP11400
      CALL MXRNP(SPIV,1,KKA,3)                                          POP11410
      CALL MXRNP(SLGV,1,KKA,3)                                          POP11420
C                                                                       POP11430
      KKS=KKS+20                                                        POP11440
 2000 KKA=KK-KKS+20                                                     POP11450
      SPIV(KKA)=PINDEX                                                  POP11460
C     GENERATE THE LENGTH OF THE GRADIENT VECTOR AND STORE IT           POP11470
      DA=0.0                                                            POP11480
      DO 2110 I=1,NPAR                                                  POP11490
 2110 DA=DA+GRADV(I)*GRADV(I)                                           POP11500
      SLGV(KKA)=SQRT(DA)                                                POP11510
C     GENERATE THE MAXIMUM ELEMENTS OF THE PARAMETER AND GRADIENT VECTORPOP11520
C     SET NIC=NIM+1 TO TERMINATE RUN IF SENSE SWITCH 3 IS SET           POP11530
C     SET NHC=NHM+1 TO TERMINATE RUN IF SENSE SWITCH 3 IS SET           POP11540
      IF(SENSE SWITCH 3)2120,2300                                       POP11550
```

```
 2120 TYPE 1050                                                        POP11560
      PRINT 1050                                                       POP11570
      NIC=NIM+1                                                        POP11580
      NHC=NHM+1                                                        POP11590
 2300 RETURN                                                           POP11600
C                                                                      POP11610
 2    CONTINUE                                                         POP11620
      RETURN                                                           POP11630
C                                                                      POP11640
 3    CONTINUE                                                         POP11650
      RETURN                                                           POP11660
C                                                                      POP11670
 4    CONTINUE                                                         POP11680
      RETURN                                                           POP11690
C                                                                      POP11700
C     ITPRT INITIALIZATION                                             POP11710
 5    CONTINUE                                                         POP11720
      PRINT 1060                                                       POP11730
      CALL IRANDP(1,NITPRT,IA,IA,IA,IA,IA,IA,4)                        POP11740
      IKS=0                                                            POP11750
      KKS=20                                                           POP11760
      RETURN                                                           POP11770
C                                                                      POP11780
      END                                                              POP11790
```

```
      SUBROUTINE MINFCN(NENTRY)                                          POP11800
C                                                                        POP11810
C     SUBROUTINE TO MINIMIZE A PERFORMANCE INDEX                         POP11820
C     SPECIAL VERSION FOR OPTIMIZING THE PARAMETERS IN A DYNAMICAL SYSTEPOP11830
C                                                                        POP11840
      DIMENSION PARV(20),GRADV(20)                                       POP11850
      COMMON/BLKA/PINDEX,PISTOR,PARV,EPS,GRADV,NIC,NHC,NPAR,NIM,NHM      POP11860
C                                                                        POP11870
      DIMENSION SPARV(20),DV(20)                                         POP11880
      COMMON/BLKA1/NOPT,EPSINC,EPSDEC,ISDEC,NSDEC,SNHC,SPARV,DV          POP11890
C                                                                        POP11900
 1000 FORMAT(7E10.0)                                                     POP11910
 1001 FORMAT(7F15.6)                                                     POP11920
 1120 FORMAT(6HMINFCN)                                                   POP11930
C                                                                        POP11940
      GO TO(11,12,13),NENTRY                                             POP11950
C                                                                        POP11960
C     INPUT DATA                                                         POP11970
  11  PRINT 1120                                                         POP11980
      CALL RANDP(2)                                                      POP11990
      CALL RANDPD(3,EPS,EPSINC,EPSDEC,DA,DA,DA,DA,4)                     POP12000
      CALL IRANDP(6,NIM,NHM,NOPT,NPAR,ISDEC,NSDEC,JJ,4)                  POP12010
      EPSM=EPS                                                           POP12020
C     OPTIMIZATION ALGORITHMS                                            POP12030
C     PERFORMANCE INDEX INITIALIZATION                                   POP12040
      CALL IRANDP(1,NPAR,IA,IA,IA,IA,IA,IA,4)                            POP12050
      CALL MXRNP(PARV,1,NPAR,4)                                          POP12060
      CALL GRAD(1)                                                       POP12070
      CALL EPCHNG(1,EPS,EPSDEC,GRADV)                                    POP12080
      CALL ITPRT(5)                                                      POP12090
      CALL GRADMX(1)                                                     POP12100
      CALL MINFA(1)                                                      POP12110
      CALL AVGRAD(1)                                                     POP12120
      CALL NEWRAF(1)                                                     POP12130
      CALL CNGRAD(1)                                                     POP12140
      CALL DAVIDN(1)                                                     POP12150
      CALL POWEL(1)                                                      POP12160
      RETURN                                                             POP12170
C                                                                        POP12180
C     INITIALIZE PERFORMANCE INDEX                                       POP12190
  12  CALL PINDX(1)                                                      POP12200
      RETURN                                                             POP12210
C                                                                        POP12220
C     PERFORM MINIMIZATION                                               POP12230
  13  PISTOR=10.0**6                                                     POP12240
      NIC=-1                                                             POP12250
      NHC=0                                                              POP12260
      SNHC=-1                                                            POP12270
      EPS=EPSM                                                           POP12280
C     INITIALIZE UFV                                                     POP12290
      CALL MFCS(4)                                                       POP12300
C     INITIALIZE GRADV, GRADM                                            POP12310
      DO 900 I=1,NPAR                                                    POP12320
 900  GRADV(I)=0.0                                                       POP12330
C                                                                        POP12340
      GO TO(1,2,3,4,5,6),NOPT                                            POP12350
C                                                                        POP12360
```

```
C       METHOD OF STEEPEST DESCENT                                              POP12370
 1      GO TO 1105                                                              POP12380
C       BEGINNING OF ITERATIVE LOOP                                             POP12390
C       STORE PINDEX AND PARV                                                   POP12400
 1011   PISTOR=PINDEX                                                           POP12410
        DO 1002 I=1,NPAR                                                        POP12420
 1002   SPARV(I)=PARV(I)                                                        POP12430
C       AUXILLIARY STORAGE                                                      POP12440
        CALL PINDX(4)                                                           POP12450
        CALL GRAD(2)                                                            POP12460
C       MODIFIED STEEPEST DESCENT                                               POP12470
C       SEARCH FOR A MINIMUM ALONG THE DIRECTION OF THE GRADIENT VECTOR         POP12480
C       IF NSDEC = 2                                                            POP12490
        GO TO(1115,1121),NSDEC                                                  POP12500
 1121   CALL MINFA(2)                                                           POP12510
        GO TO 1105                                                              POP12520
 1115   CALL EPCHNG(2,EPS,EPSDEC,GRADV)                                         POP12530
        DO 1100 I=1,NPAR                                                        POP12540
 1100   PARV(I)=SPARV(I)-EPS*GRADV(I)                                           POP12550
        CALL MFCS(4)                                                            POP12560
 1105   CALL PINDX(2)                                                           POP12570
        IF(PINDEX-PISTOR)1101,1102,1102                                         POP12580
 1101   NIC=NIC+1                                                               POP12590
        IF(NHC-SNHC)1200,1200,1210                                              POP12600
 1200   EPS=EPS*EPSINC                                                          POP12610
 1210   SNHC=NHC                                                                POP12620
        CALL ITPRT(1)                                                           POP12630
        IF(NIC-NIM)1011,1110,1110                                               POP12640
 1102   NHC=NHC+1                                                               POP12650
        EPS=EPS*EPSDEC                                                          POP12660
        CALL ITPRT(1)                                                           POP12670
        IF(NHC-NHM)1115,1110,1110                                               POP12680
 1110   RETURN                                                                  POP12690
C                                                                               POP12700
C       METHOD OF AVERAGE GRADIENT                                              POP12710
 2      CONTINUE                                                                POP12720
        CALL AVGRAD(2)                                                          POP12730
        RETURN                                                                  POP12740
C                                                                               POP12750
C       METHOD OF NEWTON RAPHSON                                                POP12760
 3      CONTINUE                                                                POP12770
        CALL NEWRAF(2)                                                          POP12780
        RETURN                                                                  POP12790
C                                                                               POP12800
C       METHOD OF THE CONJUGATE GRADIENT                                        POP12810
 4      CONTINUE                                                                POP12820
        CALL CNGRAD(2)                                                          POP12830
        RETURN                                                                  POP12840
C                                                                               POP12850
C       METHOD OF DAVIDON                                                       POP12860
 5      CONTINUE                                                                POP12870
        CALL DAVIDN(2)                                                          POP12880
        RETURN                                                                  POP12890
C                                                                               POP12900
C       METHOD OF POWELL                                                        POP12910
 6      CONTINUE                                                                POP12920
        CALL POWEL(2)                                                           POP12930
        RETURN                                                                  POP12940
C                                                                               POP12950
        END                                                                     POP12960
```

```
      SUBROUTINE PINDX(NENTRY)                                          POP12970
C                                                                       POP12980
C     SUBROUTINE TO EVALUATE FIGURE PERFORMANCE                         POP12990
C     INDEX  J=XT*X                                                     POP13000
C                                                                       POP13010
      DIMENSION XV(10000),AM(3000),AIM(3000),XFV(50),UFV(50),ASCALV(50),POP13020
     1 FSCALV(50),XFSV(50),YFSV(50),XFRV(50),DUMV(50),UFAV(50),LACTV(50)POP13030
      COMMON/BLKMFC/ N,NR,NRA,XV,MODE,LACTV,MODOP,NSNSWT                POP13040
      EQUIVALENCE (XV(1),AM(1)),(XV(3001),AIM(1)),(XV(6001),XFV(1)),    POP13050
     1 (UFV(1),XV(6101)),(ASCALV(1),XV(6151)),(FSCALV(1),XV(6201)),     POP13060
     2 (XFSV(1),XV(6251)),(YFSV(1),XV(6301)),(XFRV(1),XV(6351)),        POP13070
     3 (UFAV(1),XV(6401)),(DUMV(1),XV(6451))                            POP13080
C                                                                       POP13090
      GO TO(1,2),NENTRY                                                 POP13100
C                                                                       POP13110
C     INITIALIZATION                                                    POP13120
  1   CONTINUE                                                          POP13130
      RETURN                                                            POP13140
C                                                                       POP13150
C     CALCULATE PINDEX=XFVT*XFV                                         POP13160
  2   PINDEX=0.0                                                        POP13170
      DO 2100 I=1,N                                                     POP13180
      CALL FIGSEN(2,I,XFV,XFSV,YFSV,FSCALV)                             POP13190
 2100 PINDEX=PINDEX+XFV(I)*XFV(I)                                       POP13200
      RETURN                                                            POP13210
C                                                                       POP13220
      END                                                               POP13230
```

```
DUMMY SUBROUTINE PACKAGE

      SUBROUTINE ANGRAD(NENTRY)                                         POP13240
      RETURN                                                            POP13250
      END                                                               POP13260

      SUBROUTINE AVGRAD(NENTRY)                                         POP13270
                                                                        POP13280
      RETURN                                                            POP13290
      END

      SUBROUTINE CNGRAD(NENTRY)                                         POP13300
      RETURN                                                            POP13310
      END                                                               POP13320

      SUBROUTINE DAVIDN(NENTRY)                                         POP13330
      RETURN                                                            POP13340
      END                                                               POP13350

      SUBROUTINE DESX(NENTRY)                                           POP13360
      RETURN                                                            POP13370
      END                                                               POP13380

      SUBROUTINE GRADMX(NENTRY)                                         POP13390
      RETURN                                                            POP13400
      END                                                               POP13410

      SUBROUTINE MINFA(NENTRY)                                          POP13420
      RETURN                                                            POP13430
      END                                                               POP13440

      SUBROUTINE NEWRAF(NENTRY)                                         POP13450
      RETURN                                                            POP13460
      END                                                               POP13470

      SUBROUTINE PENFCN(NENTRY)                                         POP13480
                                                                        POP13490
      RETURN                                                            POP13500
      END
      SUBROUTINE POWEL(NENTRY)                                          POP13510
      RETURN                                                            POP13520
      END                                                               POP13530
```

PRECEDING PAGE BLANK NOT FILMED

APPENDIX C
MATHEMATICAL OPERATIONS PACKAGE

C.0 Introduction

This appendix contains listings for the following programs.

```
      FUNCTION ELM(A,L,M,N)                                              MOP10000
C                                                                        MOP10010
C     FUNCTION RETURNS THE VALUE OF THE L,M TH ELEMENT OF THE MATRIX     MOP10020
C     A WHICH HAS N COLUMNS AND AN ARBITRARY NUMBER OF ROWS              MOP10030
C                                                                        MOP10040
      DIMENSION A(1)                                                     MOP10050
C                                                                        MOP10060
      ELM=A(M*N-N+L)                                                     MOP10070
      RETURN                                                             MOP10080
C                                                                        MOP10090
      END                                                                MOP10100
```

```
      SUBROUTINE ELMA(NENTRY,A,I,J,V,N)                                  MOP10110
C                                                                        MOP10120
C     SUBROUTINE TO WRITE INTO AND READ FROM MEMORY THE I,JTH ELEMENT    MOP10130
C     OF MATRIX A WHICH IS STORED IN GENERAL FORM.                       MOP10140
C                                                                        MOP10150
      DIMENSION A(1)                                                     MOP10160
C                                                                        MOP10170
      GO TO(1,2),NENTRY                                                  MOP10180
C                                                                        MOP10190
C     A(I,J)=V                                                           MOP10200
  1   A(I+(J-1)*N)=V                                                     MOP10210
      RETURN                                                             MOP10220
C                                                                        MOP10230
C     V=A(I,J)                                                           MOP10240
  2   V=A(I+(J-1)*N)                                                     MOP10250
C                                                                        MOP10260
      RETURN                                                             MOP10270
C                                                                        MOP10280
      END                                                                MOP10290
```

```
      SUBROUTINE GMADD(A,B,R,N,M)                                       MOP10300
C                                                                       MOP10310
C     SUBROUTINE PERFORMS MATRIX ADDITION, R=A+B, WHERE A,B AND R ARE   MOP10320
C     N BY M MATRICES.                                                  MOP10330
C                                                                       MOP10340
      DIMENSION A(1),B(1),R(1)                                          MOP10350
C                                                                       MOP10360
      NM=N*M                                                            MOP10370
      DO 110 I=1,NM                                                     MOP10380
  110 R(I)=A(I)+B(I)                                                    MOP10390
      RETURN                                                            MOP10400
C                                                                       MOP10410
      END                                                               MOP10420
```

```
      SUBROUTINE GMPRD(A,B,R,N,M,L)                                     MOP10430
C                                                                       MOP10440
C     FORM THE PRODUCT R=A*B WHERE A IS A N*M MATRIX AND B IS A M*L MATRMOP10450
C     A,B AND R ARE STORED IN GENERAL MATRIX FORM COLUMN BY COLUMN      MOP10460
C                                                                       MOP10470
      DIMENSION A(1),B(1),R(1)                                          MOP10480
C                                                                       MOP10490
      IR=0                                                              MOP10500
      IK=-M                                                             MOP10510
      DO 10 K=1,L                                                       MOP10520
      IK=IK+M                                                           MOP10530
      DO 10 J=1,N                                                       MOP10540
      IR=IR+1                                                           MOP10550
      JI=J-N                                                            MOP10560
      IB=IK                                                             MOP10570
      R(IR)=0.0                                                         MOP10580
      DO 10 I=1,M                                                       MOP10590
      JI=JI+N                                                           MOP10600
      IB=IB+1                                                           MOP10610
   10 R(IR)=R(IR)+A(JI)*B(IB)                                           MOP10620
      RETURN                                                            MOP10630
C                                                                       MOP10640
      END                                                               MOP10650
```

```
      SUBROUTINE GMSUB(A,B,R,N,M)                                       MOP10660
C                                                                       MOP10670
C     SUBROUTINE PERFORMS MATRIX SUBTRACTION, R=A-B, WHERE A,B AND R AREMOP10680
C     N BY M MATRICES.                                                  MOP10690
C                                                                       MOP10700
      DIMENSION A(1),B(1),R(1)                                          MOP10710
C                                                                       MOP10720
      NM=N*M                                                            MOP10730
      DO 110 I=1,NM                                                     MOP10740
  110 R(I)=A(I)-B(I)                                                    MOP10750
      RETURN                                                            MOP10760
C                                                                       MOP10770
      END                                                               MOP10780
```

```
      SUBROUTINE GMTRA(A,R,N,M)                                         MOP10790
C                                                                       MOP10800
C     TRANSPOSE A GENERAL MATRIX                                        MOP10810
C                                                                       MOP10820
C     A - NAME OF MATRIX TO BE TRANSPOSED                               MOP10830
C     R - NAME OF RESULTANT MATRIX                                      MOP10840
C     N - NUMBER OF ROWS OF A AND COLUMNS OF R                          MOP10850
C     M - NUMBER OF COLUMNS OF A AND ROWS OF R                          MOP10860
C                                                                       MOP10870
      DIMENSION A(1),R(1)                                               MOP10880
C                                                                       MOP10890
      IR=0                                                              MOP10900
      DO 10 I=1,N                                                       MOP10910
      IJ=I-N                                                            MOP10920
      DO 10 J=1,M                                                       MOP10930
      IJ=IJ+N                                                           MOP10940
      IR=IR+1                                                           MOP10950
   10 R(IR)=A(IJ)                                                       MOP10960
      RETURN                                                            MOP10970
C                                                                       MOP10980
      END                                                               MOP10990
```

```
      SUBROUTINE  GTOSYM(X,XS,NX)                                       MOP11000
C                                                                       MOP11010
C     PROGRAM CONVERTS A SQUARE NX BY NX MATRIX INTO A VECTOR WHOSE LENGMOP11020
C     NF*(NF+1)/2 AND WHOSE ELEMENTS CONSIST OF THE UPPER TRIANGLE OF   MOP11030
C     THE NX BY NX MATRIX, STORED IN COLUMNAR FORM.                     MOP11040
C     THE NX BY NX MATRIX MUST BE STORED IN A VECTOR WHOSE LENGTH IS NX*MOP11050
C     IN COLUMNAR FORM, BEFORE THIS ROUTINE IS CALLED.                  MOP11060
C                                                                       MOP11070
      DIMENSION X(1),XS(1)                                              MOP11080
C                                                                       MOP11090
      LL=0                                                              MOP11100
      DO 10 J=1,NX                                                      MOP11110
      DO 10 I=1,J                                                       MOP11120
      LL=LL+1                                                           MOP11130
      K=(J-1)*NX+I                                                      MOP11140
 10   XS(LL)=X(K)                                                       MOP11150
      RETURN                                                            MOP11160
C                                                                       MOP11170
      END                                                               MOP11180
```

```
      SUBROUTINE LOC(I,J,IR,N,M,MS)                                     MOP11190
C                                                                       MOP11200
C     SUBROUTINE TO GENERATE VECTOR SUBSCRIPT FOR AN ELEMENT IN A MATRIXMOP11210
C     OF SPECIFIED STORAGE MODE.                                        MOP11220
C     MS =0  SUBSCRIPT IS COMPUTED FOR A MATRIX WITH N*M ELEMENTS       MOP11230
C            IN STORAGE (GENERAL MATRIX)                                MOP11240
C     MS=1   SUBSCRIPT IS COMPUTED FOR A MATRIX WITH N*(N+1)/2 IN       MOP11250
C            STORAGE (UPPER TRIANGLE OF SUMMETRIC MATRIX).  IF          MOP11260
C            ELEMENT IS IN LOWER TRIANGULAR PORTION, SUBSCRIPT IS       MOP11270
C            CORRESPONDING ELEMENT IN UPPER TRIANGLE.                   MOP11280
C     MS=2   SUBSCRIPT IS COMPUTED FOR A MATRIX WITH N ELEMENTS         MOP11290
C            IN STORAGE (DIAGONAL ELEMENTS OF DIAGONAL MATRIX).         MOP11300
C            IF ELEMENT IS NOT ON DIAGONAL (AND THEREFORE NOT IN        MOP11310
C            STORAGE), IR IS SET TO ZERO.                               MOP11320
C                                                                       MOP11330
      IX=I                                                              MOP11340
      JX=J                                                              MOP11350
      IA=I-J                                                            MOP11360
      IF(MS-1) 10,20,30                                                 MOP11370
 10   IRX=N*(JX-1)+IX                                                   MOP11380
      GO TO 36                                                          MOP11390
 20   IF(IA)22,24,24                                                    MOP11400
 22   IRX=IX+(JX*JX-JX)/2                                               MOP11410
      GO TO 36                                                          MOP11420
 24   IRX=JX+(IX*IX-IX)/2                                               MOP11430
      GO TO 36                                                          MOP11440
   30 IRX=0                                                             MOP11450
      IF(IX-JX)36,32,36                                                 MOP11460
   32 IRX=IX                                                            MOP11470
 36   IR=IRX                                                            MOP11480
      RETURN                                                            MOP11490
C                                                                       MOP11500
      END                                                               MOP11510
```

```
      SUBROUTINE MCPY(A,R,N,M,MS)                                        MOP11520
C                                                                        MOP11530
C     MCPY COPIES ENTIRE N BY M MATRIX A INTO N BY M MATRIX R            MOP11540
C           MS  - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX A (AND R)  MOP11550
C                 0 - GENERAL                                            MOP11560
C                 1 - SYMMETRIC                                          MOP11570
C                 2 - DIAGONAL                                           MOP11580
C                                                                        MOP11590
      DIMENSION A(1),R(1)                                                MOP11600
C                                                                        MOP11610
C     COMPUTE VECTOR LENGTH, IT                                          MOP11620
      CALL LOC(N,M,IT,N,M,MS)                                            MOP11630
C     COPY MATRIX                                                        MOP11640
      DO 1 I=1,IT                                                        MOP11650
    1 R(I)=A(I)                                                          MOP11660
      RETURN                                                             MOP11670
C                                                                        MOP11680
      END                                                                MOP11690
```

```
      SUBROUTINE MMADD(N,ALPHA,A,BETA,B,C)                               MOP11700
C                                                                        MOP11710
C     SUBROUTINE TO FORM COMBINATION C=ALPHA*A+BETA*B                    MOP11720
C                                                                        MOP11730
      DIMENSION A(1),B(1),C(1)                                           MOP11740
C                                                                        MOP11750
      DO 1 I=1,N                                                         MOP11760
    1 C(I)=ALPHA*A(I)+BETA*B(I)                                          MOP11770
      RETURN                                                             MOP11780
C                                                                        MOP11790
      END                                                                MOP11800
```

```
      SUBROUTINE MPRD(A,B,R,N,M,MSA,MSB,L)                              MOP11810
C     MPRD MULTIPLIES N BY M MATRIX A BY M BY L MATRIX B AND STORES THE MOP11820
C     PRODUCT INTO N BY L MATRIX R                                      MOP11830
C           MSA - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX A         MOP11840
C                 0 - GENERAL                                         MOP11850
C                 1 - SYMMETRIC                                       MOP11860
C                 2 - DIAGONAL                                        MOP11870
C           MSB - SAME AS MSA EXCEPT FOR MATRIX B                     MOP11880
C                                                                     MOP11890
      DIMENSION A(1),B(1),R(1)                                        MOP11900
C                                                                     MOP11910
C     SPECIAL CASE FOR DIAGONAL BY DIAGONAL                           MOP11920
      MS=MSA*10.+MSB                                                  MOP11930
      IF(MS-22) 30,10,30                                              MOP11940
   10 DO 20 I=1,N                                                     MOP11950
   20 R(I)=A(I)*B(I)                                                  MOP11960
      RETURN                                                          MOP11970
C                                                                     MOP11980
C     ALL OTHER CASES                                                 MOP11990
   30 IR=1                                                            MOP12000
      DO 90 K=1,L                                                     MOP12010
      DO 90 J=1,N                                                     MOP12020
      R(IR)=0                                                         MOP12030
      DO 80 I=1,M                                                     MOP12040
      IF(MS)40,60,40                                                  MOP12050
   40 CALL LOC(J,I,IA,N,M,MSA)                                        MOP12060
      CALL LOC(I,K,IB,M,L,MSB)                                        MOP12070
      IF(IA)50,80,50                                                  MOP12080
   50 IF(IB)70,80,70                                                  MOP12090
   60 IA=N*(I-1)+J                                                    MOP12100
      IB=M*(K-1)+I                                                    MOP12110
   70 R(IR)=R(IR)+A(IA)*B(IB)                                         MOP12120
   80 CONTINUE                                                        MOP12130
   90 IR=IR+1                                                         MOP12140
      RETURN                                                          MOP12150
C                                                                     MOP12160
      END                                                             MOP12170
```

```
      SUBROUTINE MTRA(A,R,N,M,MS)                                       MOP12180
C                                                                       MOP12190
C     MTRA TRANSPOSES N BY M MATRIX A TO FORM M BY N MATRIX R           MOP12200
C           MS  - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX A (AND R) MOP12210
C                 0 - GENERAL                                           MOP12220
C                 1 - SYMMETRIC                                         MOP12230
C                 2 - DIAGONAL                                          MOP12240
C                                                                       MOP12250
      DIMENSION A(1),R(1)                                               MOP12260
C                                                                       MOP12270
C     IF MS IS 1 OR 2, COPY A                                           MOP12280
      IF(MS) 10,20,10                                                   MOP12290
 10   CALL MCPY(A,R,N,N,MS)                                             MOP12300
      RETURN                                                            MOP12310
C                                                                       MOP12320
C     TRANSPOSE GENERAL MATRIX                                          MOP12330
   20 IR=0                                                              MOP12340
      DO 30 I=1,N                                                       MOP12350
      IJ=I-N                                                            MOP12360
      DO 30 J=1,M                                                       MOP12370
      IJ=IJ+N                                                           MOP12380
      IR=IR+1                                                           MOP12390
 30   R(IR)=A(IJ)                                                       MOP12400
      RETURN                                                            MOP12410
C                                                                       MOP12420
      END                                                               MOP12430
```

```
      SUBROUTINE SYMTOG(XS,X,NX)                                        MOP12440
C                                                                       MOP12450
C     PROGRAM CONVERTS A SYM. MATRIX VECTOR (IN SUPPRESSED SYM. STORAGE)MOP12460
C     WHOSE LENGTH IS NX*(NX+1)/2, INTO A GENERAL MATRIX VECTOR WHOSE   MOP12470
C     LENGTH IS NX*NX.                                                  MOP12480
C                                                                       MOP12490
      DIMENSION X(1),XS(1)                                              MOP12500
C                                                                       MOP12510
      LL=0                                                              MOP12520
      DO 10 J=1,NX                                                      MOP12530
      DO 10 I=1,J                                                       MOP12540
      LL=LL+1                                                           MOP12550
      K=(J-1)*NX+I                                                      MOP12560
      M=(I-1)*NX+J                                                      MOP12570
      X(M)=XS(LL)                                                       MOP12580
 10   X(K)=XS(LL)                                                       MOP12590
      RETURN                                                            MOP12600
C                                                                       MOP12610
      END                                                               MOP12620
```

```
      SUBROUTINE SINV%N,AI,B,D<                                          MOP12630
C                                                                        MOP12640
C*****SUBROUTINE TO GENERATE THE INVERSE OF THE MATRIX AI                MOP12650
C     THE MATRICES AI AND B ARE STORED IN GENERAL FORM                   MOP12660
C     INPUT MATRIX IS AI                                                 MOP12670
C     OUTPUT INVERSE MATRIX IS B                                         MOP12680
C     N IS THE ORDER OF AI                                               MOP12690
C     D IS THE DETERMINANT OF AI                                         MOP12700
C                                                                        MOP12710
C           L - WORK VECTOR OF LENGTH N                                  MOP12720
C           M - WORK VECTOR OF LENGTH N                                  MOP12730
C                                                                        MOP12740
C           THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT    MOP12750
C           IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT     MOP12760
C           THE MATRIX IS SINGULAR.                                      MOP12770
C*****WITH MODIFICATIONS TO INPUT MATRIX IN VECTOR FORMAT                MOP12780
C                                                                        MOP12790
C                                                                        MOP12800
C                                                                        MOP12810
      DIMENSION AI%400<,B%400<,A%400<,L%20<,M%20<                        MOP12820
C                                                                        MOP12830
C        .....MOP12840
C                                                                        MOP12850
C        IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE   MOP12860
C        C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION       MOP12870
C        STATEMENT WHICH FOLLOWS.                                        MOP12880
C                                                                        MOP12890
C     DOUBLE PRECISION A,D,BIGA,HOLD                                     MOP12900
C                                                                        MOP12910
C        THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS     MOP12920
C        APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS       MOP12930
C        ROUTINE.                                                        MOP12940
C                                                                        MOP12950
C        THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO       MOP12960
C        CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  ABS IN STATEMENT   MOP12970
C        10 MUST BE CHANGED TO DABS.                                     MOP12980
C        STORAGE OF AI ELEMENT IN A                                      MOP12990
C                                                                        MOP13000
      KK#N*N                                                             MOP13010
      DO 5 J#1,KK                                                        MOP13020
    5 A%J<#AI%J<                                                         MOP13030
C                                                                        MOP13040
C        .....MOP13050
C                                                                        MOP13060
C        SEARCH FOR LARGEST ELEMENT                                      MOP13070
C                                                                        MOP13080
      D#1.0                                                              MOP13090
      NK#-N                                                              MOP13100
      DO 80 K#1,N                                                        MOP13110
      NK#NK&N                                                            MOP13120
      L%K<#K                                                             MOP13130
      M%K<#K                                                             MOP13140
      KK#NK&K                                                            MOP13150
      BIGA#A%KK<                                                         MOP13160
      DO 20 J#K,N                                                        MOP13170
      IZ#N*%J-1<                                                         MOP13180
      DO 20 I#K,N                                                        MOP13190
```

```
      IJ#IZ&I                                                          MDP13200
 10   IF%ABS%BIGA<-ABS%A%IJ<<< 15,20,20                                MDP13210
 15   BIGA#A%IJ<                                                       MDP13220
      L%K<#I                                                           MDP13230
      M%K<#J                                                           MDP13240
 20   CONTINUE                                                         MDP13250
C                                                                      MDP13260
C        INTERCHANGE ROWS                                              MDP13270
C                                                                      MDP13280
      J#L%K<                                                           MDP13290
      IF%J-K<35,35,25                                                  MDP13300
 25   KI#K-N                                                           MDP13310
      DO 30 I#1,N                                                      MDP13320
      KI#KI&N                                                          MDP13330
      HOLD#-A%KI<                                                      MDP13340
      JI#KI-K&J                                                        MDP13350
      A%KI<#A%JI<                                                      MDP13360
 30   A%JI<#HOLD                                                       MDP13370
C                                                                      MDP13380
C        INTERCHANGE COLUMNS                                           MDP13390
C                                                                      MDP13400
 35   I#M%K<                                                           MDP13410
      IF%I-K<45,45,38                                                  MDP13420
 38   JP#N*%I-1<                                                       MDP13430
      DO 40 J#1,N                                                      MDP13440
      JK#NK&J                                                          MDP13450
      JI#JP&J                                                          MDP13460
      HOLD#-A%JK<                                                      MDP13470
      A%JK<#A%JI<                                                      MDP13480
 40   A%JI<#HOLD                                                       MDP13490
C                                                                      MDP13500
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS       MDP13510
C        CONTAINED IN BIGA)                                            MDP13520
C                                                                      MDP13530
 45   IF%BIGA< 48,46,48                                                MDP13540
 46   D#0.0                                                            MDP13550
      GO TO 150                                                        MDP13560
 48   DO 55 I#1,N                                                      MDP13570
      IF%I-K<50,55,50                                                  MDP13580
 50   IK#NK&I                                                          MDP13590
      A%IK<#A%IK</%-BIGA<                                              MDP13600
 55   CONTINUE                                                         MDP13610
C                                                                      MDP13620
C        REDUCE MATRIX                                                 MDP13630
C                                                                      MDP13640
      DO 65 I#1,N                                                      MDP13650
      IK#NK&I                                                          MDP13660
      HOLD#A%IK<                                                       MDP13670
      IJ#I-N                                                           MDP13680
      DO 65 J#1,N                                                      MDP13690
      IJ#IJ&N                                                          MDP13700
      IF%I-K<60,65,60                                                  MDP13710
 60   IF%J-K<62,65,62                                                  MDP13720
 62   KJ#IJ-I&K                                                        MDP13730
      A%IJ<#HOLD*A%KJ<&A%IJ<                                           MDP13740
 65   CONTINUE                                                         MDP13750
C                                                                      MDP13760
```

```
C          DIVIDE ROW BY PIVOT                                          MDP13770
C                                                                       MDP13780
       KJ#K-N                                                           MDP13790
       DO 75 J#1,N                                                      MDP13800
       KJ#KJ+N                                                          MDP13810
       IF%J-K<70,75,70                                                  MDP13820
  70   A%KJ<#A%KJ</BIGA                                                 MDP13830
  75   CONTINUE                                                         MDP13840
C                                                                       MDP13850
C          PRODUCT OF PIVOTS                                            MDP13860
C                                                                       MDP13870
       D#D*BIGA                                                         MDP13880
C                                                                       MDP13890
C          REPLACE PIVOT BY RECIPROCAL                                  MDP13900
C                                                                       MDP13910
       A%KK<#1.0/BIGA                                                   MDP13920
  80   CONTINUE                                                         MDP13930
C                                                                       MDP13940
C          FINAL ROW AND COLUMN INTERCHANGE                             MDP13950
C                                                                       MDP13960
       K#N                                                              MDP13970
  100  K#%K-1<                                                          MDP13980
       IF%K< 150,150,105                                                MDP13990
  105  I#L%K<                                                           MDP14000
       IF%I-K<120,120,108                                               MDP14010
  108  JO#N*%K-1<                                                       MDP14020
       JR#N*%I-1<                                                       MDP14030
       DO 110 J#1,N                                                     MDP14040
       JK#JO+J                                                          MDP14050
       HOLD#A%JK<                                                       MDP14060
       JI#JR+J                                                          MDP14070
       A%JK<#-A%JI<                                                     MDP14080
  110  A%JI<#HOLD                                                       MDP14090
  120  J#M%K<                                                           MDP14100
       IF%J-K< 100,100,125                                              MDP14110
  125  KI#K-N                                                           MDP14120
       DO 130 I#1,N                                                     MDP14130
       KI#KI+N                                                          MDP14140
       HOLD#A%KI<                                                       MDP14150
       JI#KI-K+J                                                        MDP14160
       A%KI<#-A%JI<                                                     MDP14170
  130  A%JI<#HOLD                                                       MDP14180
       GO TO 100                                                        MDP14190
  150  LL#0                                                             MDP14200
C                                                                       MDP14210
       KK#N*N                                                           MDP14220
       DO 151 J#1,KK                                                    MDP14230
  151  B%J<#A%J<                                                        MDP14240
       RETURN                                                           MDP14250
C                                                                       MDP14260
       END                                                              MDP14270
```

PRECEDING PAGE BLANK NOT FILMED

APPENDIX D
INPUT OUTPUT OPERATIONS PACKAGE

D.0 Introduction

This appendix contains listings for the following programs.

```
      SUBROUTINE IMXRNP(M,NA,NB,NENTRY)                                 IOP10000
C                                                                       IOP10010
C     SUBROUTINE READS,PRINTS AND STORES INTEGER NA*NB MATRIX           IOP10020
C     MATRIX IS STORED IN GENERAL FORM COLUMN BY COLUMN                 IOP10030
C                                                                       IOP10040
      DIMENSION M(1)                                                    IOP10050
C                                                                       IOP10060
 1000 FORMAT(7I10)                                                      IOP10070
 1002 FORMAT(7I15)                                                      IOP10080
 1003 FORMAT(7F10.4)                                                    IOP10090
C                                                                       IOP10100
C     READ IN NA BY NB MATRIX ROW-WISE AND STORE INTO 1 DIMENSION       IOP10110
C     VECTOR COLUMN-WISE.                                               IOP10120
      GO TO(1,1,2,4,2),NENTRY                                           IOP10130
C                                                                       IOP10140
 1    J=NA*NB-NA+1                                                      IOP10150
      DO 15 I=1,NA                                                      IOP10160
      READ 1000, (M(K),K=I,J,NA)                                        IOP10170
 15   J=J+1                                                             IOP10180
C                                                                       IOP10190
      GO TO(2,3,3,2),NENTRY                                             IOP10200
C                                                                       IOP10210
C     PRINT NA BY NB MATRIX ROW-WISE                                    IOP10220
 2    CONTINUE                                                          IOP10230
      JJ=NA*NB-NA+1                                                     IOP10240
      DO 11 II=1,NA                                                     IOP10250
      IF(NENTRY-5)12,10,12                                              IOP10260
 10   PUNCH 1003,(M(L),L=II,JJ,NA)                                      IOP10270
      GO TO 11                                                          IOP10280
 12   PRINT 1002, (M(L),L=II,JJ,NA)                                     IOP10290
 11   JJ=JJ+1                                                           IOP10300
C                                                                       IOP10310
 3    RETURN                                                            IOP10320
C                                                                       IOP10330
C     READ AND PRINT HEADING CARD BEFORE READING AND PRINTING MATRIX    IOP10340
 4    CALL RANDP(4)                                                     IOP10350
      GO TO 1                                                           IOP10360
C                                                                       IOP10370
      END                                                               IOP10380
```

```
      SUBROUTINE IRANDP(ND,IA,IB,IC,ID,IE,IF,IG,NENTRY)                 IOP10390
C                                                                       IOP10400
C     SUBROUTINE TO READ AND PRINT INTEGER DATA                         IOP10410
C                                                                       IOP10420
      DIMENSION IV(7)                                                   IOP10430
C                                                                       IOP10440
 1000 FORMAT(7I10)                                                      IOP10450
 1010 FORMAT(7I15)                                                      IOP10460
C                                                                       IOP10470
      GO TO(1,2,1,4),NENTRY                                             IOP10480
 1    READ 1000,IA,IB,IC,ID,IE,IF,IG                                    IOP10490
      GO TO(2,2,3,2),NENTRY                                             IOP10500
 2    IV(1)=IA                                                          IOP10510
      IV(2)=IB                                                          IOP10520
      IV(3)=IC                                                          IOP10530
      IV(4)=ID                                                          IOP10540
      IV(5)=IE                                                          IOP10550
      IV(6)=IF                                                          IOP10560
      IV(7)=IG                                                          IOP10570
      PRINT 1010,(IV(I),I=1,ND)                                         IOP10580
 3    RETURN                                                            IOP10590
C                                                                       IOP10600
 4    CALL RANDP(4)                                                     IOP10610
      GO TO 1                                                           IOP10620
C                                                                       IOP10630
      END                                                               IOP10640
```

```
      SUBROUTINE MXRNP(VA,NA,NB,NENTRY)                                 IOP10650
C                                                                       IOP10660
C     SUBROUTINE READS AND/OR PRINTS THE NA*NB MATRIX VA WHICH IS STOREDIOP10670
C     GENERAL FORM COLUMN BY COLUMN                                     IOP10680
C                                                                       IOP10690
      DIMENSION VA(1)                                                   IOP10700
C                                                                       IOP10710
 1000 FORMAT(7E10.0)                                                    IOP10720
 1002 FORMAT(7F15.6)                                                    IOP10730
 1003 FORMAT(7I10)                                                      IOP10740
C                                                                       IOP10750
      GO TO(1,1,2,4,2),NENTRY                                           IOP10760
C                                                                       IOP10770
C     READ IN NA BY NB MATRIX ROW-WISE AND STORE INTO 1 DIMENSION       IOP10780
C     VECTOR COLUMN-WISE.                                               IOP10790
  1   J=NA*NB-NA+1                                                      IOP10800
      DO 15 I=1,NA                                                      IOP10810
      READ 1000, (VA(K),K=I,J,NA)                                       IOP10820
 15   J=J+1                                                             IOP10830
      GO TO(2,3,3,2),NENTRY                                             IOP10840
C                                                                       IOP10850
  2   CONTINUE                                                          IOP10860
C     PRINT NA BY NB MATRIX ROW-WISE                                    IOP10870
      JJ=NA*NB-NA+1                                                     IOP10880
      DO 11 II=1,NA                                                     IOP10890
      IF(NENTRY-5)12,10,12                                              IOP10900
 10   PUNCH 1003,(VA(L),L=II,JJ,NA)                                     IOP10910
      GO TO 11                                                          IOP10920
 12   PRINT 1002, (VA(L),L=II,JJ,NA)                                    IOP10930
 11   JJ=JJ+1                                                           IOP10940
      RETURN                                                            IOP10950
C                                                                       IOP10960
  3   RETURN                                                            IOP10970
C                                                                       IOP10980
C     READ AND PRINT HEADING CARD BEFORE READING AND PRINTING MATRIX    IOP10990
  4   CALL RANDP(4)                                                     IOP11000
      GO TO 1                                                           IOP11010
C                                                                       IOP11020
      END                                                               IOP11030
```

```
      SUBROUTINE NAMRNP(M,NA,NB,NENTRY)                                 IOP11040
C                                                                       IOP11050
C     SUBROUTINE READS,PRINTS AND STORES INTEGER NA*NB MATRIX           IOP11060
C     OF FOUR CHARACTER NAMES                                           IOP11070
C     MATRIX IS STORED IN GENERAL FORM COLUMN BY COLUMN                 IOP11080
C                                                                       IOP11090
      DIMENSION M(1)                                                    IOP11100
C                                                                       IOP11110
 1000 FORMAT(1X,A4,1X,A4,1X,A4,1X,A4,1X,A4,1X,A4,1X,A4,1X,A4,1X,A4,1X,A4,IOP11120
     1 1X,A4,1X,A4,1X,A4,1X,A4,1X,A4,1X,A4)                             IOP11130
 1002 FORMAT(11X,A4,11X,A4,11X,A4,11X,A4,11X,A4,11X,A4,11X,A4)          IOP11140
C                                                                       IOP11150
      GO TO(1,1,2,4),NENTRY                                             IOP11160
C                                                                       IOP11170
C     READ IN NA BY NB MATRIX ROW-WISE AND STORE INTO 1 DIMENSION       IOP11180
C     VECTOR COLUMN-WISE.                                               IOP11190
  1   J=NA*NB-NA+1                                                      IOP11200
      DO 15 I=1,NA                                                      IOP11210
      READ 1000, (M(K),K=I,J,NA)                                        IOP11220
 15   J=J+1                                                             IOP11230
      GO TO(2,3,3,2),NENTRY                                             IOP11240
C                                                                       IOP11250
C     PRINT NA BY NB MATRIX ROW-WISE                                    IOP11260
  2   CONTINUE                                                          IOP11270
      JJ=NA*NB-NA+1                                                     IOP11280
      DO 11 II=1,NA                                                     IOP11290
      PRINT 1002,(M(L),L=II,JJ,NA)                                      IOP11300
 11   JJ=JJ+1                                                           IOP11310
C                                                                       IOP11320
  3   RETURN                                                            IOP11330
C                                                                       IOP11340
C     READ IN HEADING CARD BEFORE READING AND PRINTIND M                IOP11350
  4   CALL RANDP(4)                                                     IOP11360
      GO TO 1                                                           IOP11370
C                                                                       IOP11380
      END                                                               IOP11390
```

```
      SUBROUTINE RANDP(NENTRY)                                          IOP11400
C                                                                       IOP11410
C     SUBROUTINE TO READ AND PRINT HEADING CARDS                        IOP11420
C                                                                       IOP11430
      DIMENSION FNAME(8)                                                IOP11440
      DOUBLE PRECISION FNAME                                            IOP11450
C                                                                       IOP11460
 1000 FORMAT(8A8)                                                       IOP11470
 1001 FORMAT(1H1)                                                       IOP11480
 1010 FORMAT(2X,A8,2X,A8,2X,A8,2X,A8,2X,A8,2X,A8,2X,A8,2X,A8)           IOP11490
 1020 FORMAT(7X,A8,7X,A8,7X,A8,7X,A8,7X,A8,7X,A8,7X,A8,7X,A8)           IOP11500
C                                                                       IOP11510
      GO TO(2000,2010,2020,2030),NENTRY                                 IOP11520
C                                                                       IOP11530
 2000 PRINT 1001                                                        IOP11540
 2010 READ 1000,(FNAME(I),I=1,8)                                        IOP11550
      PRINT 1000,(FNAME(I),I=1,8)                                       IOP11560
      RETURN                                                            IOP11570
C                                                                       IOP11580
 2020 PRINT 1001                                                        IOP11590
 2030 READ 1010,(FNAME(I),I=1,7)                                        IOP11600
      PRINT 1020,(FNAME(I),I=1,7)                                       IOP11610
      RETURN                                                            IOP11620
C                                                                       IOP11630
      END                                                               IOP11640
```

```
      SUBROUTINE RANDPD(ND,DA,DB,DC,DD,DE,DF,DG,NENTRY)                 IOP11650
C                                                                       IOP11660
C     SUBROUTINE TO READ AND PRINT FLOATING POINT DATA                  IOP11670
C                                                                       IOP11680
      DIMENSION DV(7)                                                   IOP11690
C                                                                       IOP11700
 1000 FORMAT(7E10.0)                                                    IOP11710
 1010 FORMAT(7F15.6)                                                    IOP11720
C                                                                       IOP11730
      GO TO(1,2,1,4),NENTRY                                             IOP11740
C                                                                       IOP11750
 1    READ 1000,DA,DB,DC,DD,DE,DF,DG                                    IOP11760
      GO TO(2,2,3,2),NENTRY                                             IOP11770
 2    DV(1)=DA                                                          IOP11780
      DV(2)=DB                                                          IOP11790
      DV(3)=DC                                                          IOP11800
      DV(4)=DD                                                          IOP11810
      DV(5)=DE                                                          IOP11820
      DV(6)=DF                                                          IOP11830
      DV(7)=DG                                                          IOP11840
      PRINT 1010,(DV(I),I=1,ND)                                         IOP11850
 3    RETURN                                                            IOP11860
C                                                                       IOP11870
 4    CALL RANDP(4)                                                     IOP11880
      GO TO 1                                                           IOP11890
C                                                                       IOP11900
      END                                                               IOP11910
```

REFERENCES

1. MacKinnon, D., P. Madden, and P. Farrell, Optical Mirror Figure Control, MIT Charles Stark Draper Laboratory Report R-665, Cambridge, Massachusetts, May 1970.

2. MacKinnon, D., "Active Control of Primary Mirror Figure," NASA SP-233 (Proceedings, NASA Workshop on Optical Telescope Technology, MSFC, Huntsville, Alabama, April 29 - May 1, 1969).

3. Active Optical System for Spaceborne Telescopes, NASA CR-66297, Perkin-Elmer Corporation, Norwalk, Connecticut, October 14, 1966.

4. Active Optical System for Spaceborne Telescopes, Vol II, NASA CR-66489, Perkin-Elmer Corporation, Norwalk, Connecticut, December 7, 1967.

5. Development of an Active Optics Concept Using a Thin Deformable Mirror, Final Report, Perkin-Elmer Corporation, Norwalk, Connecticut, 1968.

6. Robertson, Hugh J., Development of an Active Optics Concept Using a Thin Deformable Mirror, NASA CR-1593, Perkin-Elmer Corporation, Norwalk, Connecticut, August 1970.

7. Crane, R., Advanced Figure Sensor, Final Report, NASA Electronics Research Center, September 1969.

8. Watt, G.J., "Actuators for Active Optics," NASA SP-233 (Proceedings, NASA Workshop on Optical Telescope Technology, MSFC, Huntsville, Alabama, April 29 - May 1, 1969).

9. XDS Sigma 5/7 Extended Fortran IV Reference Manual. El Segundo: Xerox Data Systems, April 1970.

10. XDS Fortran II Reference Manual. El Segundo: Xerox Data Systems, February 1967.

11. XDS Sigma 5/7 Symbol/Meta-Symbol Reference Manual. El Segundo: Xerox Data Systems, December 1969.

12. PDP-15 Descriptive Literature. Maynard: The Digital Equipment Corporation.

13. PDP-15 Price List. Maynard: The Digital Equipment Corporation, July 7, 1969.